# Fuzzy Metric Clustering and Dynamic Programming

岩村覚三
(城西大 数学科)

堀口正之
(東京電機大 情報環境学部)

蔵野正美
(千葉大 教育学部)

2004年3月4日
全92頁

# Fuzzy Metric Clustering and Dynamic Programming の全体構成

a. データ総数Nと分割数Mを入力する。Nの上限はNMAX, Mの上限はMMAXとする。States情報を各々のstateに付き1ワード=4バイト使ってメモリ1GB〜2GBのパソコンでソフトが動作するようにNMAX=25とする。

b. D.P Networkを作成する。

c. $\alpha=0.01$からきざみ幅$0.01=\frac{1}{100}$で$\alpha$を増加させ$\alpha=1.0$になるまで以下を行なう。各$\alpha$ ($=0.01, 0.02, \cdots, 1.00$)に対しFuzzy Metric Clusteringから決まる$d_{ij}^{(\alpha)}$ ($1 \leqq i, j \leqq N$, $i \neq j$)を用いて多段目的関数最小化DP問題を解く。
$\alpha$に依存して最適partitionと、その時の目的関数を決める。各$\alpha$に対する最適partitionとその時の目的関数値は2次元配列に持ち計算終了時にファイル BstPrttn.txt に出力する。
結果の出力は各$\alpha$に対し、毎回 BstPrttn.txt に出力してもよい。その時は2次元配列は不要になる。

参考資料


[1] 岩村覚三、堀口正之、堀池真琴、ダイナミックプログラミングを用いたアジイメトリッククラスタリング、数理解析研究所講究録1630(2009年2月)77−88

[2] R.E. Jensen, A dynamic programming algorithm for cluster analysis, Oper. Res. 17(1969) 1034−1057

# Distribution forms の作り方

例 $N=7, M=3$.

③ 1 1 1

④ 2 1 1 　// 1 2 1 や 1 1 2 は不要、作らない。

⑤ 3 1 1 　2 2 1 　// 2 1 2 は不要。

⑥ 4 1 1 　~~3 2 1~~ 　// 3 1 2 , 2 3 1 は作らない。
3 2 1 　~~2 3 1~~ 　// 2 2 1 から作る 3 2 1 は
~~3 1 2~~ 　2 2 2 　// 重複により捨てる。
4 1 1 　3 2 1 　2 2 2 　//

⑦ 5 1 1 　~~4 2 1~~ 　~~3 2 2~~ 　// ――― で消されたもの
4 2 1 　3 3 1 　~~2 3 2~~ 　// は不当 distribution form
~~4 1 2~~ 　3 2 2 　~~2 2 3~~ 　// を示し、――― で消された
// ものは重複 distribution
5 1 1 　4 2 1 　3 3 1 　3 2 2 // form を示す。

例 $N=6, M=3$

⑥ 4 1 1 　3 2 1 　2 2 2

$\{4\}\ \{1\}\ \{1\}$
$\{3\}\ \{2\}\ \{1\}$
$\{2\}\ \{2\}\ \{2\}$

$max(1)=4$, $max(2)=5$, $max(3)=6$
$min(1)=2$, $min(2)=4$, $min(3)=6$

$$NS(1)=\binom{6}{2}+\binom{6}{3}+\binom{6}{4}=\frac{6\cdot5}{2}+\frac{6\cdot5\cdot4}{3\cdot2}+\frac{6\cdot5}{2}=15+20+15=50$$

$max(2)-min(1)=5-2=3$, $K=1$, $min(K+1)=4$, $max(K+1)=5$,

$$TA(1)=\sum_{L=2}^{4}\sum_{J=1}^{3}FA(L,J)$$

2003.12.5−II ②

# M=2, N=4の時の distribution forms

$M_0=2$ as $N=4 \geqq 4=2M$.
with NO Reduction.

1 1

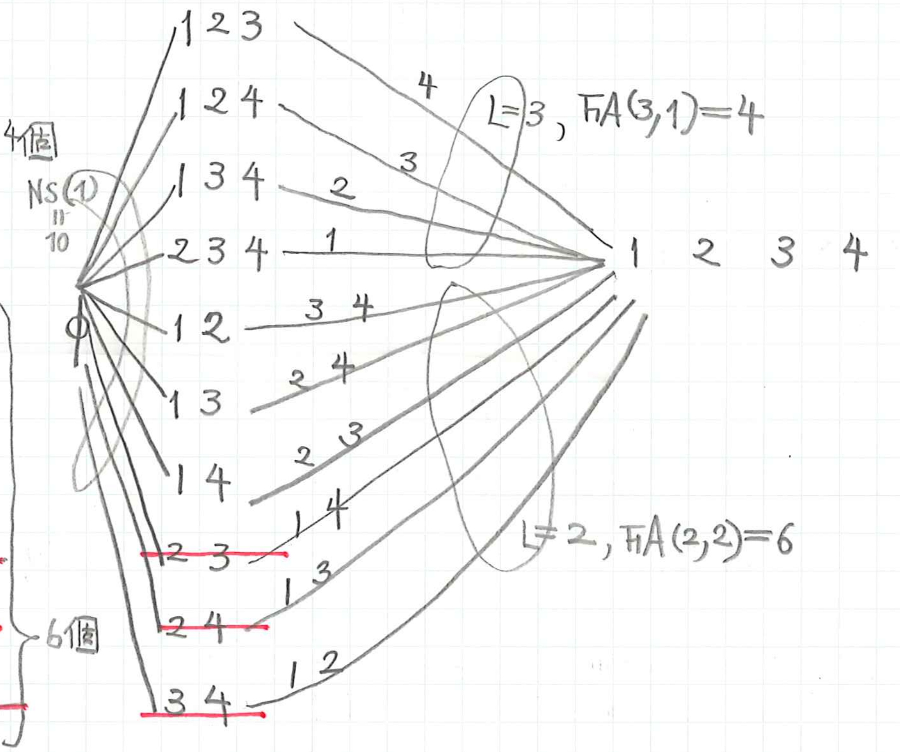
2 1 // 1 2 は不要

3 1 2 2

2つの distribution forms {3}{1}, {2}{2} がある。

$$\left.\begin{array}{ll}(1\ 2\ 3) & (4)\\ (1\ 2\ 4) & (3)\\ (1\ 3\ 4) & (2)\\ (2\ 3\ 4) & (1)\end{array}\right\} 4\text{個}$$

$$\left.\begin{array}{ll}(1\ 2) & (3\ 4)\\ (1\ 3) & (2\ 4)\\ (1\ 4) & (2\ 3)\\ \cancel{(2\ 3)} & \cancel{(1\ 4)}\\ \cancel{(2\ 4)} & \cancel{(1\ 3)}\\ \cancel{(3\ 4)} & \cancel{(1\ 2)}\end{array}\right\} 6\text{個}$$




Stage0 Stage1 Stage2

max(1)=4−2+1=3
min(1)=1(4/2)=2

max(2)=4−2+2=4
min(2)=2(4/2)=4

$NS(0)=1$

$$NS(1)=\sum_{L=2}^{3}\binom{4}{L}=\binom{4}{2}+\binom{4}{3}=\frac{4\cdot 3}{2}+4=6+4=10.$$

$$NS(2)=\sum_{L=4}^{4}\binom{4}{L}=\binom{4}{4}=1,\quad max(2)-min(1)=4-2=2$$

TFA (Total Number of Feasible Arcs)

$$\Leftarrow NS(1)+\sum_{K=1}^{1}TA(1),\quad min(2)=4 \leqq L+J \leqq max(2)=4$$

$$K=1:\ TA(1)=\sum_{L=2}^{3}\sum_{J=1}^{2}FA(L,J)=\overbrace{FA(2,1)+FA(2,2)}^{4\leqq 2+J\leqq 4}+\overbrace{FA(3,1)+FA(3,2)}^{4\leqq 3+J\leqq 4}=6+4=10.$$

$FA(2,1)=0$

$FA(2,2)=\binom{4}{2}\binom{4-2}{2}=\binom{4}{2}\binom{2}{2}=\frac{4\cdot 3}{2}=6$ //

$FA(3,1)=\binom{4}{3}\binom{4-3}{1}=\binom{4}{1}=4$ //

$FA(3,2)=0$

TFA = 10+10 = 20 //

実際は14arcsで十分。□

③ 2003.12.5 — III

M=3, N=5の時の distribution forms とその network $M_0=5-3=2$ as $N=5<2M$.

with NO Reduction.

③ 1 1 1

④ 2 1 1 // 1 2 1 , 1 1 2 は不要

⑤ 3 1 1 2 2 1 // 2 1 2 は不要

2つの distribution forms {3}{1}{1} と {2}{2}{1} がある。

| | | |
|---|---|---|
| 1 2 3 | 4 | 5 |
| 1 2 4 | 3 | 5 |
| 1 2 5 | 3 | 4 |
| 1 3 4 | 2 | 5 |
| 1 3 5 | 2 | 4 |
| 1 4 5 | 2 | 3 |
| 2 3 4 | 1 | 5 |
| 2 3 5 | 1 | 4 |
| 2 4 5 | 1 | 3 |
| 3 4 5 | 1 | 2 |

10通り

| | | | |
|---|---|---|---|
| | 1 2 | 3 4 | 5 |
| | 1 2 | 3 5 | 4 |
| | 1 2 | 4 5 | 3 |
| | 1 3 | 2 4 | 5 |
| | 1 3 | 2 5 | 4 |
| | 1 3 | 4 5 | 2 |
| | 1 4 | 2 3 | 5 |
| | 1 4 | 2 5 | 3 |
| | 1 4 | 3 5 | 2 |
| | 1 5 | 2 3 | 4 |
| | 1 5 | 2 4 | 3 |
| | 1 5 | 3 4 | 2 |
| | ~~2 3~~ | ~~1 4~~ | ~~5~~ |
| | ~~2 3~~ | ~~1 5~~ | ~~4~~ |
| 生 | 2 3 | 4 5 | 1 |
| | ~~2 4~~ | ~~1 3~~ | ~~5~~ |
| | ~~2 4~~ | ~~1 5~~ | ~~3~~ |
| 生 | 2 4 | 3 5 | 1 |
| | ~~2 5~~ | ~~1 3~~ | ~~4~~ |
| | ~~2 5~~ | ~~1 4~~ | ~~3~~ |
| 生 | 2 5 | 3 4 | 1 |
| | ~~3 4~~ | ~~1 2~~ | ~~5~~ |
| | ~~3 4~~ | ~~1 5~~ | ~~2~~ |
| | ~~3 4~~ | ~~2 5~~ | ~~1~~ |
| | ~~3 5~~ | ~~1 2~~ | ~~4~~ |
| | ~~3 5~~ | ~~1 4~~ | ~~2~~ |
| | ~~3 5~~ | ~~2 4~~ | ~~1~~ |
| | ~~4 5~~ | ~~1 2~~ | ~~3~~ |
| | ~~4 5~~ | ~~1 3~~ | ~~2~~ |
| | ~~4 5~~ | ~~2 3~~ | ~~1~~ |

25通り

15通り

stage

| | | | |
|---|---|---|---|
| 0 | 1 | 2 | 3 ← K |
| max(K) | 3 | 4 | 5 |
| min(K) | 2 | 4 | 5 |

$\max(1)=5-3+1=3$, $\max(2)=5-3+2=4$, $\max(3)=5-3+3=5$

$N-M[N/M]=5-3[5/3]=5-3=2$

$\min(1)=([N/M]+1)\cdot 1$, $\min(2)=(2)\cdot 2$, $\min(3)=N-(M-3)[N/M]$

$=(2)\cdot 1=2$, $=4$, $=5-0\cdot 1=5$

by (9), (10), (11).

$NS(0)=1$,

$$NS(1)=\sum_{L=2}^{3}\binom{N}{L}=\binom{5}{2}+\binom{5}{3}=\binom{5}{2}+\binom{5}{2}=\frac{5\cdot 4}{2}+\frac{5\cdot 4}{2}=10+10=20,$$

$$NS(2)=\sum_{L=4}^{4}\binom{N}{L}=\binom{5}{4}=\binom{5}{1}=5,$$

$$NS(3)=\sum_{L=5}^{5}\binom{5}{L}=\binom{5}{5}=1.$$

$\max(2)-\min(1)=4-2=2$,

$\max(3)-\min(2)=5-4=1$,

$$TFA=NS(1)+\sum_{K=1}^{1}TA(K)=NS(1)+TA(1),$$

$$K=1:\ TA(1)=\sum_{L=2}^{3}\sum_{J=1}^{2}FA(L,J)\ \left[\underset{\overset{\shortparallel}{4}}{\min(2)}\leqq L+J\leqq \max(2)=4\right]$$

$$=FA(2,1)+FA(2,2)+FA(3,1)+FA(3,2)=50.$$

$4\leqq 2+1\leqq 4$ NO $\to$ 0,

$4\leqq 2+2\leqq 4$ $\to$ $\binom{5}{2}\binom{3}{2}=\frac{5\cdot 4}{2}\cdot 3=30$, $=10\cdot 3$

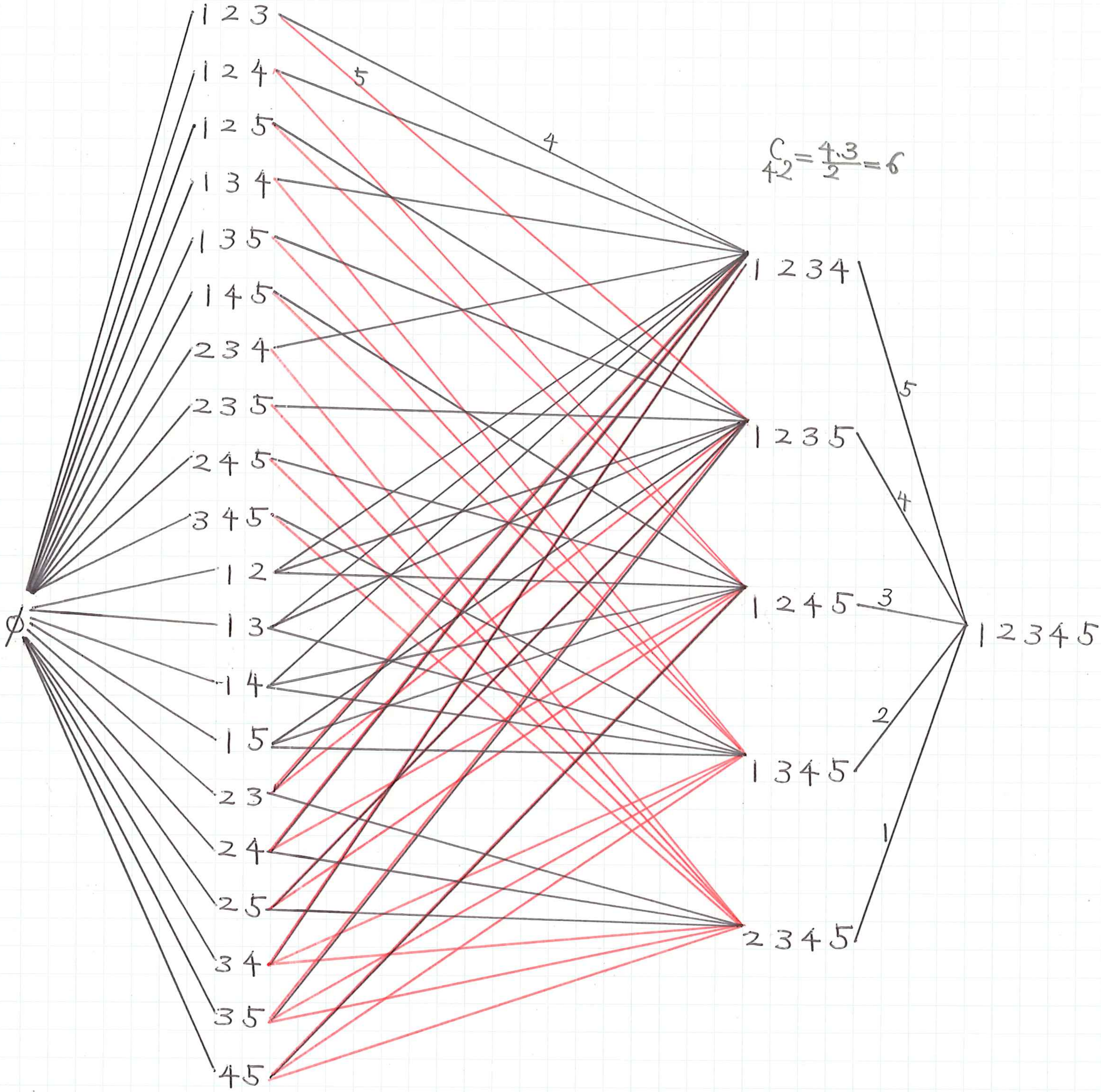
$4\leqq 3+1\leqq 4$ $\to$ $\binom{5}{3}\binom{2}{1}=10\cdot 2$, $\shortparallel$ $\frac{5\cdot 4}{2!}\cdot 2=20$,

$4\leqq 3+2\leqq 4$ NO $\to$ 0

$$TFA=20+30+20=70 \ /\!/ \quad \square$$

④ 2003, 12, 5 − IV

$M=3,\ N=5$:




Stage 0 $NS(0)=1$

Stage 1 $NS(1)=20$

Stage 2 $NS(2)=5$

Stage 3 $NS(3)=1$

# of arcs 20

# of arcs 30 +20

# of arcs 5

$30 = 15 + 15$

$20 = 10 + 10$ ?

$K=2:\ TA(2) = \sum_{L=4}^{4} \sum_{J=1}^{1} FA(L,J) = FA(4,1) = \binom{5}{4}\binom{1}{1} = \binom{5}{1} = 5.$

$5 = \min(3) \leqq 4+1 \leqq \max(3) = 5$ ? Yes.

正しいTFA $= 20 + (30+20) + 5 = 75$ //, 内 25 arcsは redundant, 50で十分? □

# $M=3,\ N=6$ の時のdistribution forms とその時の network, $M_0=3$ as $N=6 \geqq 6=2M$.

with NO Reduction

⑤

③ 1 1 1
④ 2 1 1 // 121, 112 は不要
⑤ 3 1 1  2 2 1 // 212は不要
⑥ 4 1 1  3 2 1  ~~3 2 1~~  2 2 2 // 312, 231, は不要

3つの distribution forms {4}{1}{1}, {3}{2}{1}, {2}{2}{2} がある。

$S(6,3)=90$, なぜならば

| | | |
|---|---|---|
| 1234 | 5 | 6 |
| 1235 | 4 | 6 |
| 1236 | 4 | 5 |
| 1245 | 3 | 6 |
| 1246 | 3 | 5 |
| 1256 | 3 | 4 |
| 1345 | 2 | 6 |
| 1346 | 2 | 5 |
| 1356 | 2 | 4 |
| 1456 | 2 | 3 |
| 2345 | 1 | 6 |
| 2346 | 1 | 5 |
| 2356 | 1 | 4 |
| 2456 | 1 | 3 |
| 3456 | 1 | 2 |

15通り
{4}{1}{1}

| | | | | | |
|---|---|---|---|---|---|
| 123 | 45 | 6 | 234 | 15 | 6 |
| 123 | 46 | 5 | 234 | 16 | 5 |
| 123 | 56 | 4 | 234 | 56 | 1 |
| 124 | 35 | 6 | 235 | 14 | 6 |
| 124 | 36 | 5 | 235 | 16 | 4 |
| 124 | 56 | 3 | 235 | 46 | 1 |
| 125 | 34 | 6 | 236 | 14 | 5 |
| 125 | 36 | 4 | 236 | 15 | 4 |
| 125 | 46 | 3 | 236 | 45 | 1 |
| 126 | 34 | 5 | 245 | 13 | 6 |
| 126 | 35 | 4 | 245 | 16 | 3 |
| 126 | 45 | 3 | 245 | 36 | 1 |
| 134 | 25 | 6 | 246 | 13 | 5 |
| 134 | 26 | 5 | 246 | 15 | 3 |
| 134 | 56 | 2 | 246 | 35 | 1 |
| 135 | 24 | 6 | 256 | 13 | 4 |
| 135 | 26 | 4 | 256 | 14 | 3 |
| 135 | 46 | 2 | 256 | 34 | 1 |
| 136 | 24 | 5 | 345 | 12 | 6 |
| 136 | 25 | 4 | 345 | 16 | 2 |
| 136 | 45 | 2 | 345 | 26 | 1 |
| 145 | 23 | 6 | 346 | 12 | 5 |
| 145 | 26 | 3 | 346 | 15 | 2 |
| 145 | 36 | 2 | 346 | 25 | 1 |
| 146 | 23 | 5 | 356 | 12 | 4 |
| 146 | 25 | 3 | 356 | 14 | 2 |
| 146 | 35 | 2 | 356 | 24 | 1 |
| 156 | 23 | 4 | 456 | 12 | 3 |
| 156 | 24 | 3 | 456 | 13 | 2 |
| 156 | 34 | 2 | 456 | 23 | 1 |

20

$\underbrace{\qquad\qquad}_{60通り}$

{3}{2}{1}

$_6C_3 = \frac{6\cdot5\cdot4}{3\cdot2} = 20$

| | | |
|---|---|---|
| 12 | 34 | 56 |
| 12 | 35 | 46 |
| 12 | 36 | 45 |
| 13 | 24 | 56 |
| 13 | 25 | 46 |
| 13 | 26 | 45 |
| 14 | 23 | 56 |
| 14 | 25 | 36 |
| 14 | 26 | 35 |
| 15 | 23 | 46 |
| 15 | 24 | 36 |
| 15 | 26 | 34 |
| 16 | 23 | 45 |
| 16 | 24 | 35 |
| 16 | 25 | 34 |

C

⑤ {2}{2}{2}
15通り

15+60+15=90通り //

ダメ $\frac{6!}{2!2!2!} \times \frac{1}{2!2!2!} = \frac{6\cdot5\cdot4\cdot3\cdot2}{2\cdot2\cdot2\cdot2\cdot2\cdot2}$

$\frac{6!}{2!2!2!} \times \frac{1}{3!} = \frac{6\cdot5\cdot4\cdot3\cdot2}{2\cdot2\cdot2\cdot6} = 15$ //

$_6C_4 = {_6C_2} = \frac{6\cdot5}{2} = 15$

distribution forms からも

$\max(1)=4$, $\max(2)=5=3+2$ (// $4+1$), $\max(3)=4+1+1=6$ (// $3+2+1$, $2+2+2$)

$\min(1)=2$, $\min(2)=4=2+2$, $\min(3)=6$

であることが解る。

$M=3, N=6, M_0=3$:

$max(1)=6-3+1=4$, $max(2)=6-3+2=5$, $max(3)=6-3+3=6$

$min(1)=1(6/3)=2$, $min(2)=2(6/3)=4$, $min(3)=3(6/3)=6$,

$max(2)-min(1)=5-2=3$, $max(3)-min(2)=6-4=2$.

$NS(0)=1$,

$$NS(1)=\sum_{L=2}^{4}\binom{6}{L}=\binom{6}{2}+\binom{6}{3}+\binom{6}{4}=\binom{6}{2}+\binom{6}{3}+\binom{6}{2}=\frac{6\cdot5}{2}+\frac{6\cdot5\cdot4}{3\cdot2}+\frac{6\cdot5}{2}$$

$$=15+20+15=50,$$

$$NS(2)=\sum_{L=4}^{5}\binom{6}{L}=\binom{6}{4}+\binom{6}{5}=\binom{6}{2}+\binom{6}{1}=\frac{6\cdot5}{2}+6=15+6=21,$$

$$NS(3)=\sum_{L=6}^{6}\binom{6}{L}=\binom{6}{6}=1.$$

$$TFA=NS(1)+\sum_{K=1}^{2}TA(K)=NS(1)+TA(1)+TA(2),$$

$$K=1:\ TA(1)=\sum_{L=2}^{4}\sum_{J=1}^{3}FA(L,J)\ \left[\ min(2)=4\leqq L+J\leqq 5=max(2)\ \right]$$

$$=\underset{0}{FA(2,1)}+\underset{\binom{6}{2}\binom{4}{2}}{FA(2,2)}+\underset{\binom{6}{2}\binom{4}{3}}{FA(2,3)}+\underset{\binom{6}{3}\binom{3}{1}}{FA(3,1)}+\underset{\binom{6}{3}\binom{3}{2}}{FA(3,2)}+\underset{0}{FA(3,3)}$$

$$+\underset{\binom{6}{2}\cdot2\ \to\ \binom{6}{4}\binom{2}{1}}{FA(4,1)}+\underset{0}{FA(4,2)}+\underset{0}{FA(4,3)}$$

$$=\left(\frac{6\cdot5}{2}\right)\left(\frac{4\cdot3}{2}\right)+\left(\frac{6\cdot5}{2}\right)(4)+\left(\frac{6\cdot5\cdot4}{3\cdot2}\right)(3)+\left(\frac{6\cdot5\cdot4}{3\cdot2}\right)(3)$$

$$+\left(\frac{6\cdot5}{2}\right)(2)=(15)(6)+(15)(4)+(20)(3)+(20)(3)+(15)(2)$$

$$=90+60+60+60+30=90+180+30=300\ /\!/,$$

$$K=2:\ TA(2)=\sum_{L=4}^{5}\sum_{J=1}^{2}FA(L,J)\ \left[\ min(3)=6\leqq L+J\leqq 6=max(3)\ \right]$$

$$=\underset{0}{FA(4,1)}+\underset{\binom{6}{4}\binom{2}{2}}{FA(4,2)}+\underset{\binom{6}{5}\binom{1}{1}}{FA(5,1)}+\underset{0}{FA(5,2)}=\binom{6}{2}\cdot1+\binom{6}{1}\cdot1=\frac{6\cdot5}{2}+6$$

$$=15+6=21\ /\!/,$$

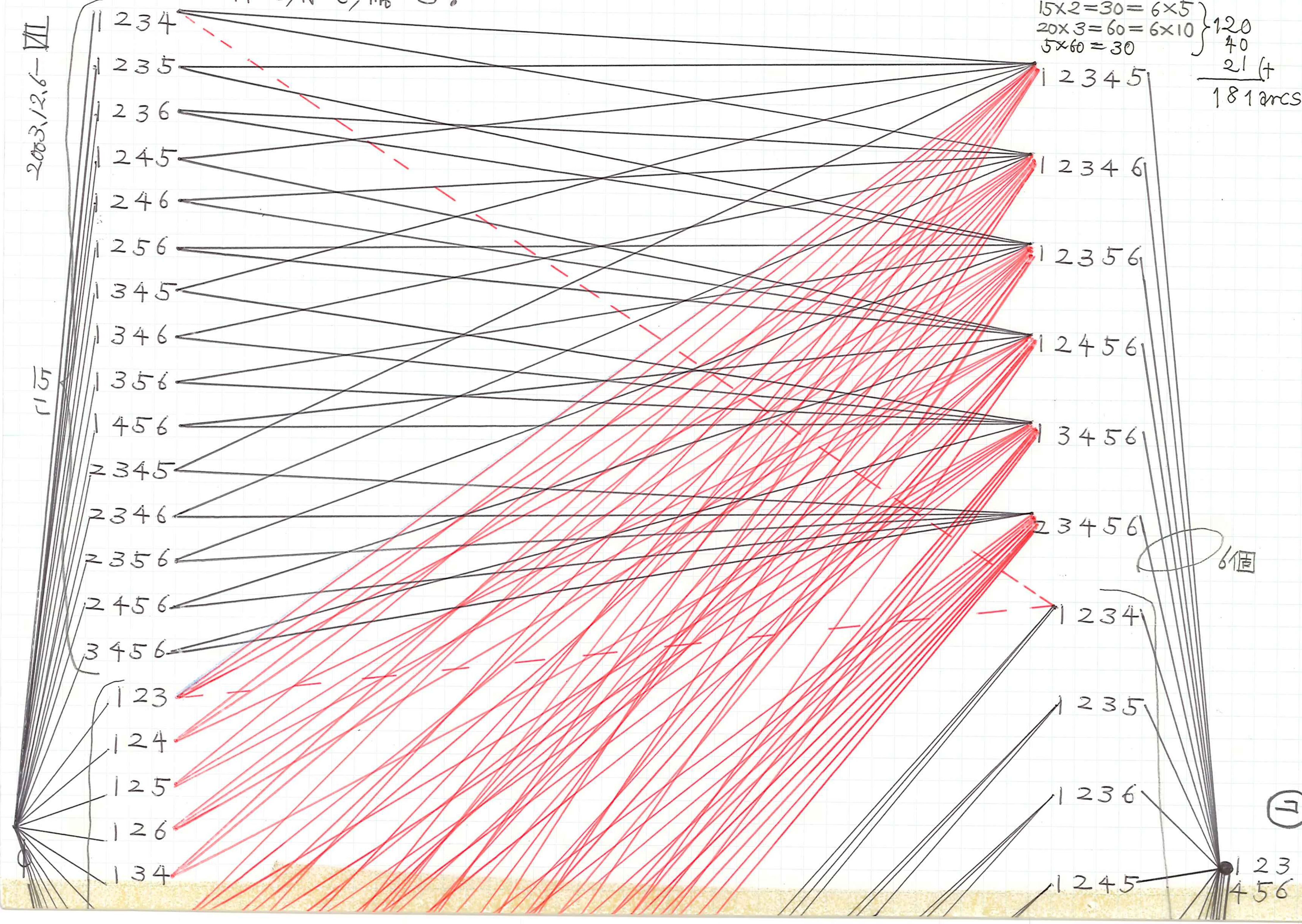
$$TFA=50+300+21=371\ /\!/.$$

2003.12.6 — VII

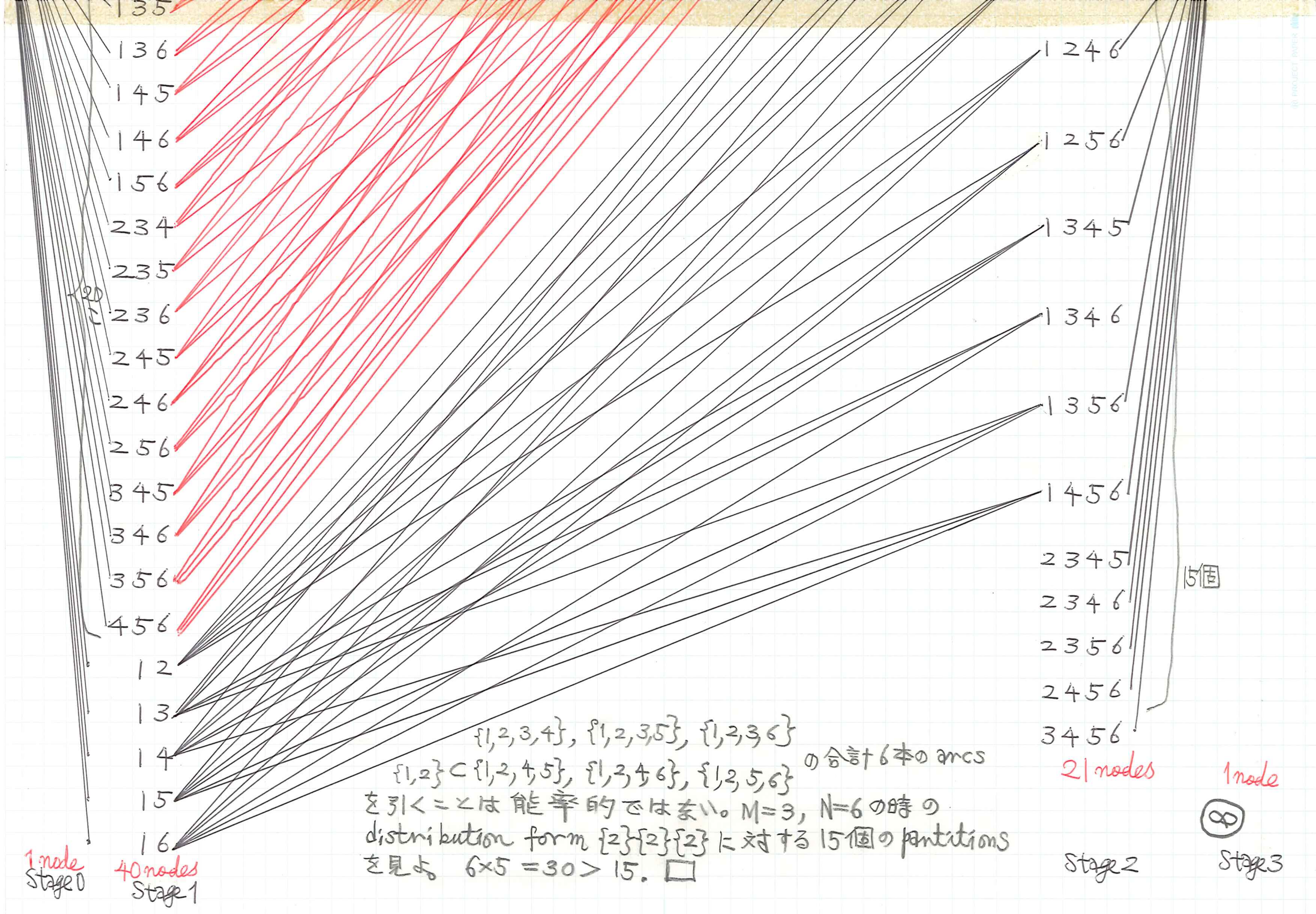
$M=3, N=6, M_0=3$:

$15\times 2 = 30 = 6\times 5$
$20\times 3 = 60 = 6\times 10$
$5\times 6 = 30$

$120 + 40 + 21 = 181$ arcs




7




{1,2,3,4}, {1,2,3,5}, {1,2,3,6}

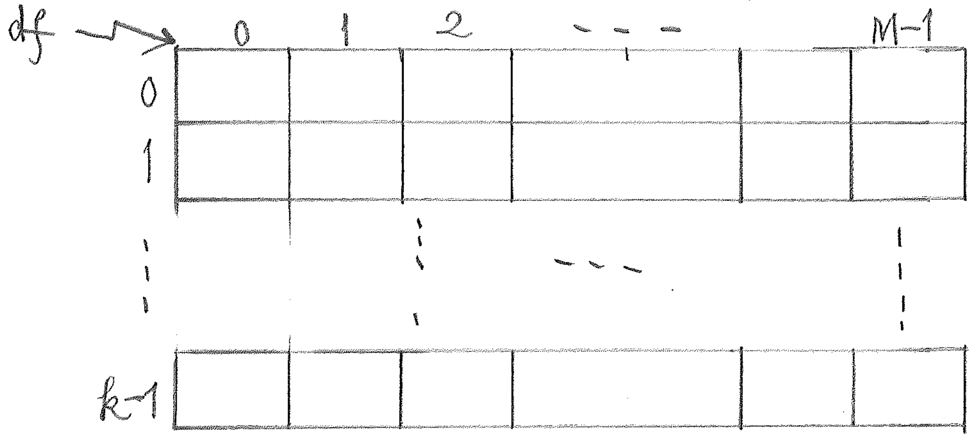
{1,2} ⊂ {1,2,4,5}, {1,2,4,6}, {1,2,5,6} の合計6本の arcs

を引くことは能率的ではない。M=3, N=6 の時の distribution form {2}{2}{2} に対する15個の partitions を見よ。 $6 \times 5 = 30 > 15$. □

8

# 固定したMに対し，N個(N≧M)の要素をM個に分割する分割 ⑨
# パターン distribution forms を全て列挙させるアルゴリズム

大きさ M の一次元配列 df[ ] を k個作るには



注：df は distribution forms から取った。

とある2次元配列 df[k][M] を準備すればよい。
理論では 1, 2, …, x となっている自然数がC言語でプログラムを作るので ここでは 0, 1, …, (x−1) と呼ぶことに注意せよ。従って N個の元 1, 2, …, N は N個の元 0, 1, …, (N−1) と呼ぶ。以下、同様とする。

a. 最初は自明な1個の distribution form (d.f.)

| | 0 | 1 | 2 | | | M−1 |
|---|---|---|---|---|---|---|
| 1−1=0 | 1 | 1 | 1 | .... | 1 | 1 |

から始める。初期値設定：$n \leftarrow M$, $k \leftarrow 1$, $nk \leftarrow 0$.

b. 要素数 n に対し，k個の distribution forms があったとする。
要素数 n+1 に対する すべての distribution forms を作る。
"Distribution forms の作り方" 例 N=7, M=3 にあるように
新しく作った distribution form が既に作成済のものと同一のものとなる時は、この distribution form は n+1

のものとして新たに追加はしない。全ての可能なdistribution formはpdf(potential distribution form)に作成する。

| | 0 | 1 | 2 | ... | M-2 | M-1 |
|---|---|---|---|---|---|---|
| pdf | $pdf_0$ | $pdf_1$ | $pdf_2$ | ... | $pdf_{M-2}$ | $pdf_{M-1}$ |

pdfが作成済のものと同一でない時のみ，これをndf(new distribution form)に登録する。このようなものを何個作ったかはnk(new k)で制御する。nkの初期値は0とする。従ってndfを完成したら$nk \leftarrow nk+1$，$k \leftarrow nk$とする。

| ndf Z→ | 0 | 1 | 2 | | | M-1 |
|---|---|---|---|---|---|---|
| 0 | $n+1-m$ | 1 | 1 | | | 1 |
| 1 | $ndf_{1,0}$ | $ndf_{1,1}$ | $ndf_{1,2}$ | | | $ndf_{1,M-1}$ |
| ⋮ | | ⋮ | | - - - - | | ⋮ |
| nk | $ndf_{nk,0}$ | $ndf_{nk,1}$ | $ndf_{nk,2}$ | | | $ndf_{nk,M-1}$ |

← (nk+1)個目のnew distribution form

例 M=3, N=5での全てのdistribution formsは

3 1 1　　2 2 1

の2つのみである。3 1 1より 4 1 1 と 3 2 1 の2つの正しいdistribution formsが作られる。一方 2 2 1 より 3 2 1 と 2 2 2 の2つの正しいdistribution formsが作られるが 3 2 1 は既に前で作成済なのでこの 3 2 1 は登録しない □

c. distribution forms $df_j$ $(0 \leqq j \leqq M-1)$ は

$$df_0 \geqq df_1 \geqq \cdots \geqq df_{M-1}$$

を満足しなければならない。これを破る distribution form は 不当(invalid) distribution form と呼ぶ。不当 distribution は +1 した position $j$ $(1 \leqq j \leqq M-1)$ に対し

$$df_{j-1} \geqq df_j \geqq df_{j+1}$$

の成立をチェックすればよい。この式が不成立の時は不当 distribution form になる。前の df[ ][i] から pdf を position $j$ で作成した時，不当 distribution form になった時は $j < j'$ を position $j'$ に +1 して新たな pdf を作る必要はない。間違い!! 不当でない distribution form を 正当(valid) distribution form と呼ぶ。

d. 要素数(n+1)に対する全ての distribution forms 作成が完了したら

$$k \leftarrow nk + 1$$

として，nk 個の new distribution forms ndf を distribution forms df に上書き move し，要素数(n+1)に対する全ての distribution forms をファイル AtDFi.txt (all the distribution forms) と display(標準出力装置)に M，n+1，k の値と共に出力する。空白行の制御を行なって異なった n に対する distribution forms がすぐに分るように出力する。$n \leftarrow n+1$ とし上述 b. から d. をくり返す。$n > N$ になったら計算終了する。

注意１：与えられた M, N (M≦N) に対し、distribution forms が何個作られるかは前以って解らないので df[M][k] と ndf[M][nk] の k, nk の値として（プログラム論理部分のサイズ）＋（OS のサイズ）と使用パソコンとのメモリ容量差以内に収まるように十分大きな値を取っておくとよい。k, nk ≦ kMAX の範囲で計算する。

注意２：M, N (M≦N) に対する distribution forms の個数を nodf(M, N) [<u>N</u>umber <u>O</u>f <u>D</u>istribution <u>F</u>orms) として M, N に関する漸化式が導ける。
$S(N, M) = nodf(M, N)$ として

$$S(N, M) = S(N-1, M-1) + M*S(N-1, M),$$

境界条件 $S(N, 1) = S(N, N) = 1 \quad (2 \leq N)$

を満たす。$1 \leq M \leq N \leq 48$ の範囲での nodf(M, N) の最大値は M=11, N=48 のとき 12,866 である。

注意３：d. における出力は n=N の時だけでよいが、debug の必要上、プログラム完全完成までは M≦n≦N な全ての n に対し AtDF.txt と display への出力を行なう。

# データ部の定義

| 変数または配列名 | データの型、配列の大きさ | 意味 |
|---|---|---|
| MMAX | int, global | Maximum of M. |
| NMAX | int, global | Maximum of N. |
| kMAX | int, global | Maximum of k and nk. |
| M | int, local | 入力データの要素数, M ≦ MMAX, M個に分割する。 |
| N | int, local | 入力データN, N個の異なるものをM個に分割する。N ≦ NMAX。 |
| j | int, local | 制御変数. |
| df | I*s, local [kMAX][MMAX] | 2次元配列 df (distribution forms), short␣int なら s＝2, int なら s＝4。 0 ≦ k ≦ kMAX－1. |
| n | int | n個の異なった要素をM個に分割, M ≦ n ≦ N。 |
| pdf | I*s [MMAX] | 一次元配列 pdf (potential distribution form), sについては df の意味を見よ。 |
| ndf | I*s, local [kMAX][MMAX] | 2次元配列 ndf (new distribution forms), 0 ≦ nk ≦ kMAX－1. |
| k | int | 現在の異なった df の個数, 1 ≦ k ≦ kMAX. |
| nk | int | 現在迄の異なった ndf の個数, 1 ≦ nk ≦ kMAX. |
| kk | int | 局所制御変数 |
| jj | int | |
| m | int | |
| k3 | int | |

注：short␣int 型つまり, I*2 の時, 表現可能な整数の範囲は
－32768 から 32767 まで。

# 必要メモリ数の計算

a. $4 \cdot 10 + kMAX \cdot MMAX \cdot s + MMAX \cdot s + kMAX \cdot MMAX \cdot s$

$= 40 + 2s \cdot kMAX \cdot MMAX + s \cdot MMAX$ (バイト)

b. MMAX＝22の時は上述a.より

$40 + 2s \cdot kMAX \cdot 22 + 22 \cdot s$ (バイト)

更に s＝4の時は

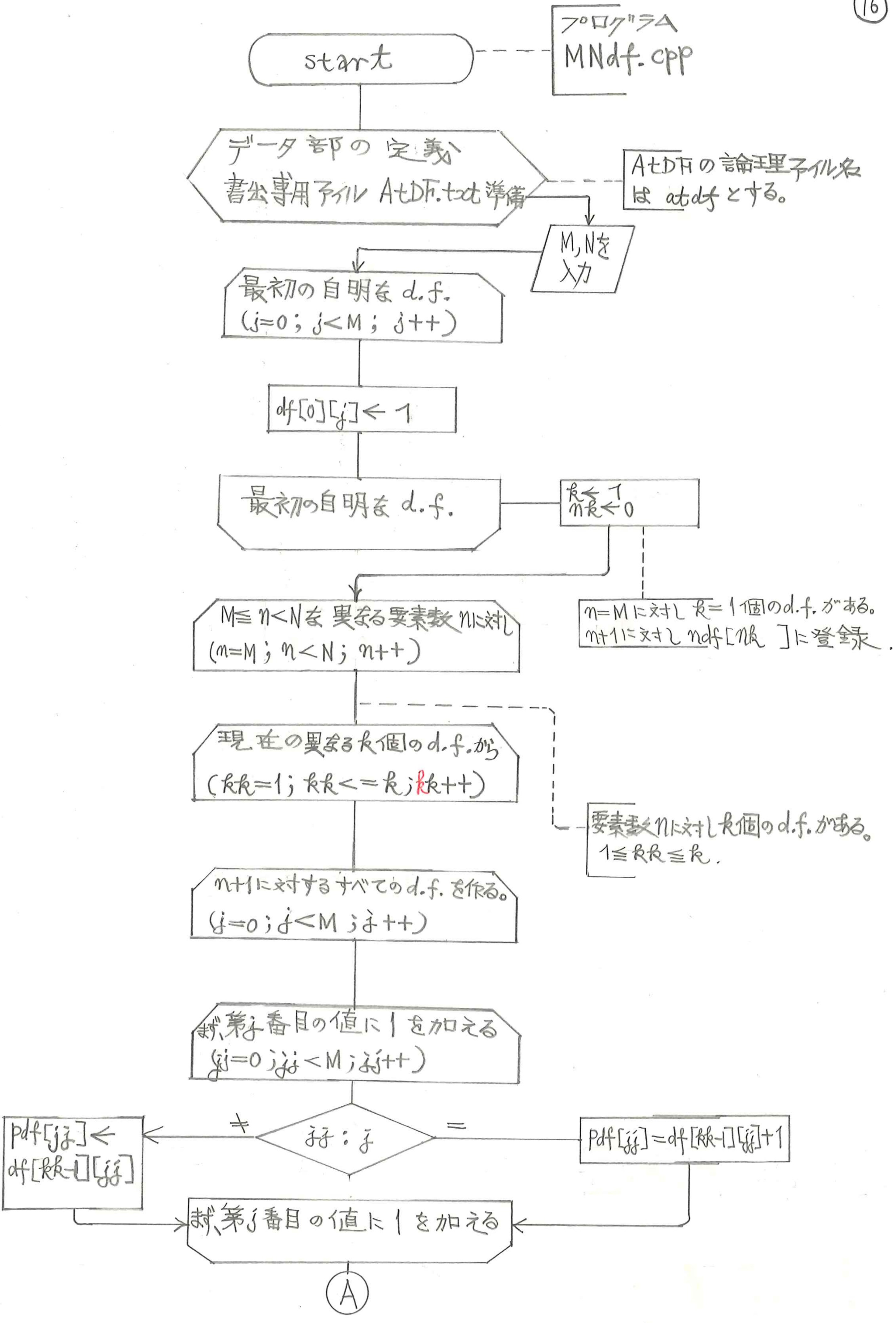
$40 + 8 \cdot 22 \cdot kMAX + 88 = 128 + 176 \cdot kMAX$

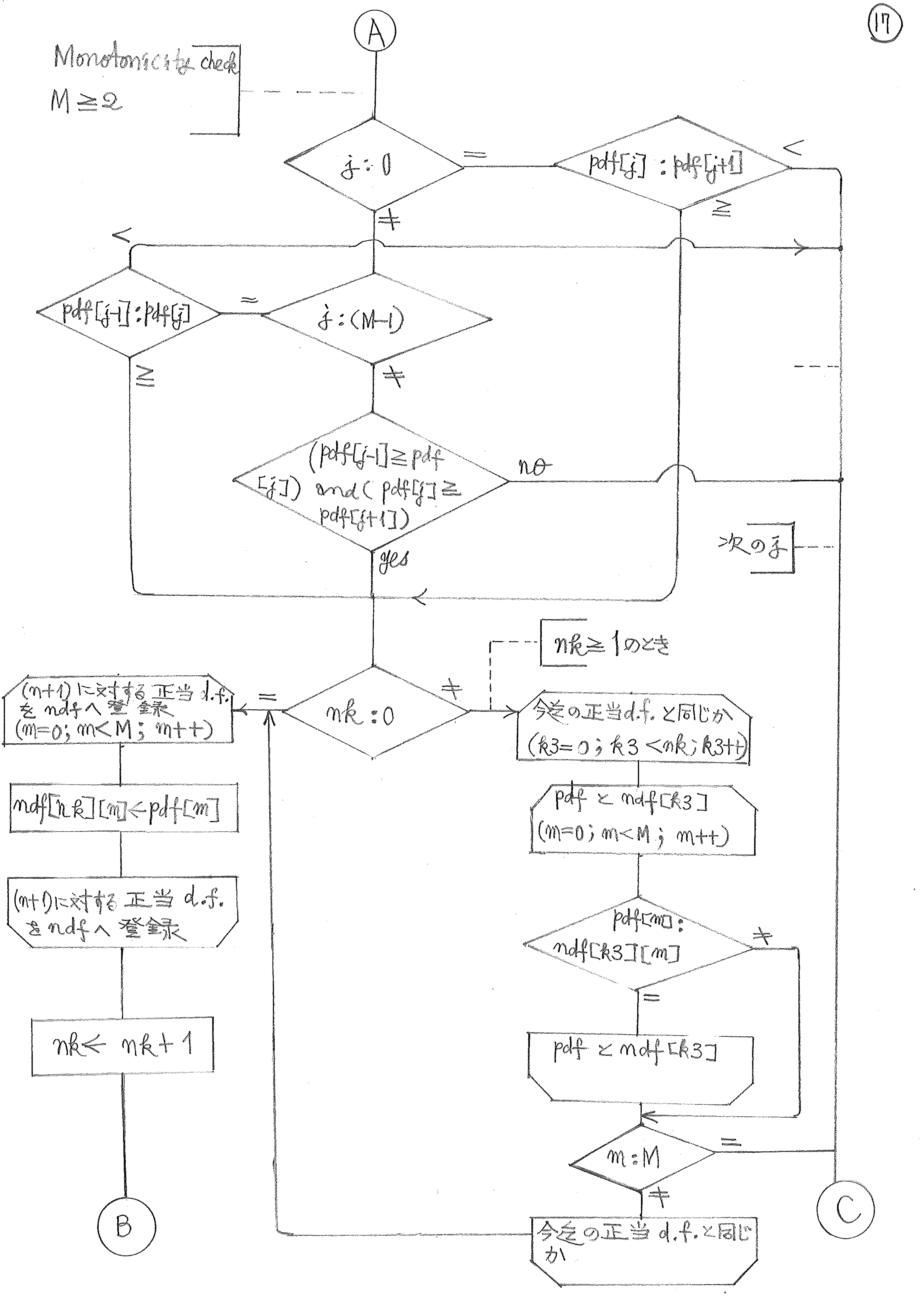
M=6, N=24のとき R.E. Jensen, Oper. Res. 17 1038より
199 unique distribution formsを持つというから、この時
$kMAX \leqq 300$と考えられる。この時は

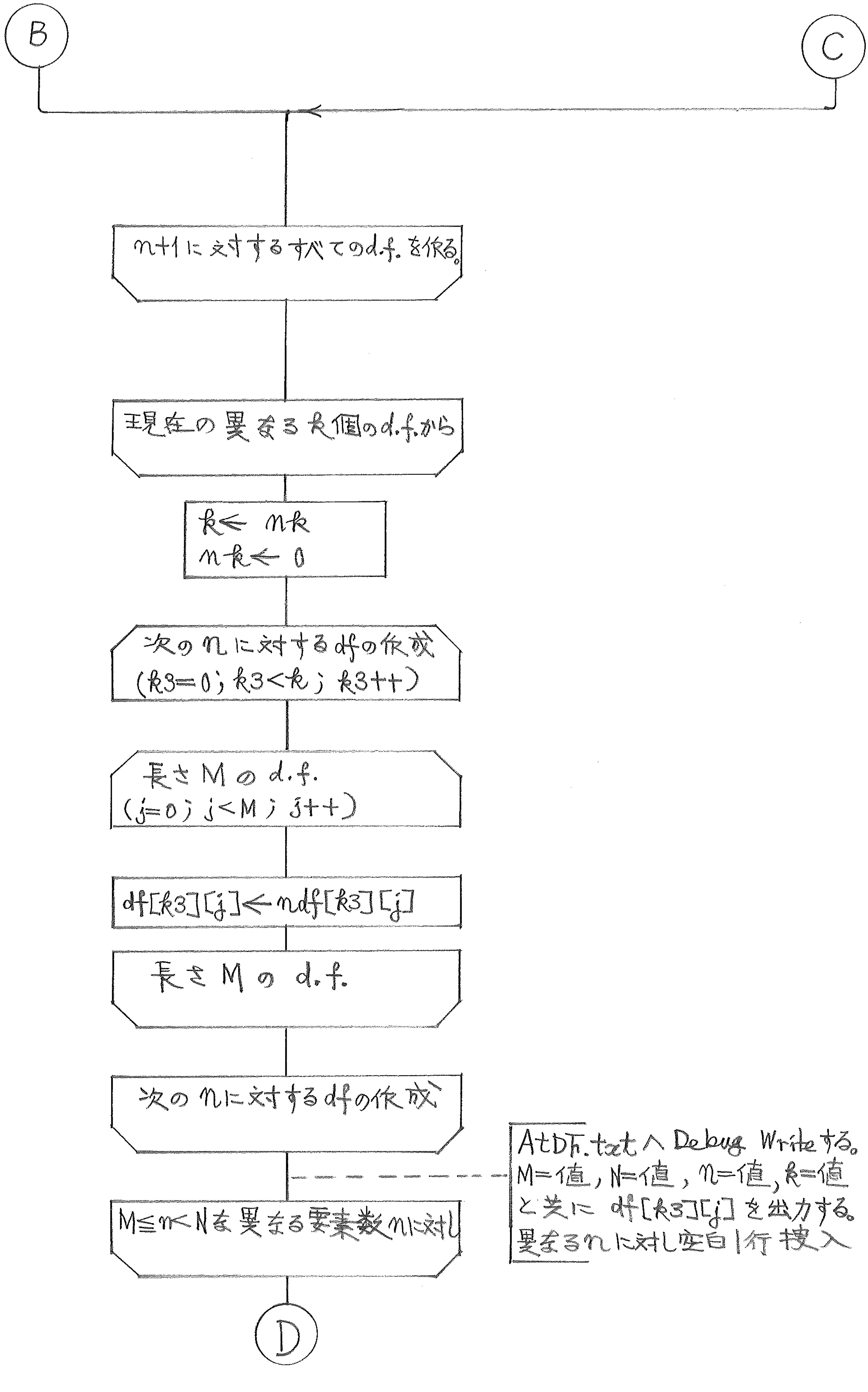
$128 + 176 \cdot 300 = 54$Kバイト＝54KB

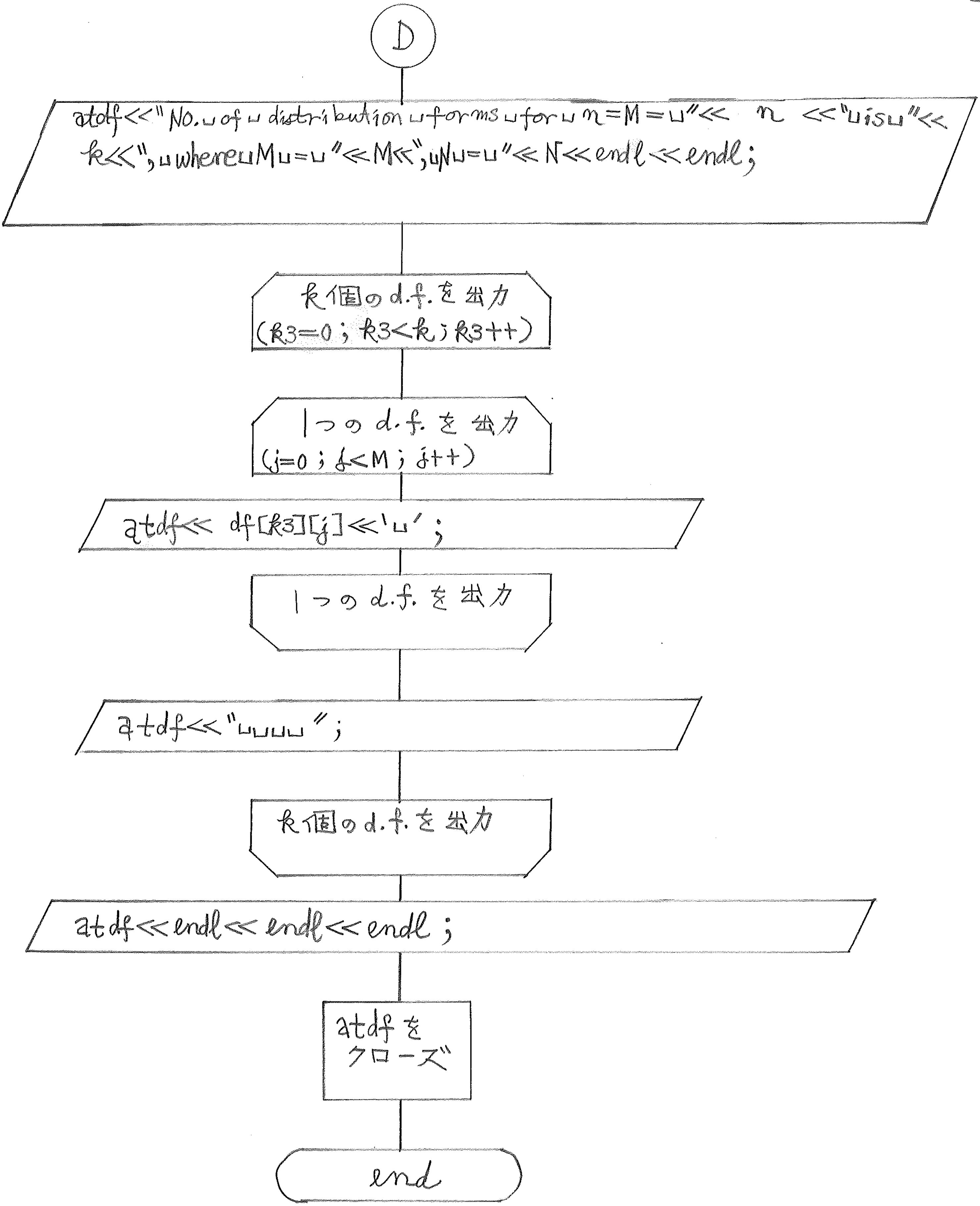
$$\begin{array}{r} 176 \\ \times)\ 300 \\ \hline 52800 \\ +)\ 128 \\ \hline 53928 \approx 54\text{KB} \end{array}$$

与えられた自然数 M(≦MMAX)と N(N≦NMAX)に対し、全ての distribution forms を求める C プログラムへのフローチャート MNdf


プログラム
MNdf.cpp
start
データ部の定義
書出専用ファイル AtDF.txt 準備
AtDFの論理ファイル名はatdfとする。
M,Nを入力
最初の自明をd.f.
(j=0; j<M; j++)
df[0][j]←1
最初の自明をd.f.
k←1
nk←0
n=Mに対しk=1個のd.f.がある。
n+1に対しndf[nk ]に登録.
M≦n<Nを異なる要素数nに対し
(n=M; n<N; n++)
現在の異なるk個のd.f.から
(kk=1; kk<=k; kk++)
要素数nに対しk個のd.f.がある。
1≦kk≦k.
n+1に対するすべてのd.f.を作る。
(j=0; j<M; j++)
まず、第j番目の値に1を加える
(jj=0; jj<M; jj++)
jj : j
≠
=
pdf[jj]←
df[kk-1][jj]
pdf[jj]=df[kk-1][jj]+1
まず、第j番目の値に1を加える
A


A
Monotonicity check
M ≧ 2
j : 0
=
pdf[j] : pdf[j+1]
<
≧
≠
<
pdf[j-1] : pdf[j]
=
j : (M-1)
≧
≠
(pdf[j-1] ≧ pdf[j]) and (pdf[j] ≧ pdf[j+1])
no
yes
次の子
nk ≧ 1のとき
(m+1)に対する正当 d.f. を ndf へ登録 (m=0; m<M; m++)
=
nk : 0
≠
今迄の正当d.f.と同じか (k3=0 ; k3 <nk ; k3++)
ndf[nk][m] ← pdf[m]
pdf と ndf[k3] (m=0 ; m<M ; m++)
(m+1)に対する正当 d.f. を ndf へ登録
pdf[m] : ndf[k3][m]
≠
=
nk ← nk + 1
pdf と ndf[k3]
m : M
=
≠
B
C
今迄の正当 d.f. と同じか


B
C
n+1に対するすべてのd.f.を作る。
現在の異なるk個のd.f.から
k← nk
nk← 0
次のnに対するdfの作成
(k3=0; k3<k; k3++)
長さMのd.f.
(j=0; j<M; j++)
df[k3][j]← ndf[k3][j]
長さMのd.f.
次のnに対するdfの作成
AtDF.txtへDebug Writeする。
M=値, N=値, n=値, k=値
と共に df[k3][j]を出力する。
異なるnに対し空白1行挿入
M≦n<Nを異なる要素数nに対し
D


D
atdf<<"No.␣of␣distribution␣forms␣for␣n=M=␣"<< n <<"␣is␣"<<
k<<",␣where␣M␣=␣"<<M<<",␣N␣=␣"<<N<<endl<<endl;
k個のd.f.を出力
(k3=0; k3<k; k3++)
1つのd.f.を出力
(j=0; j<M; j++)
atdf<< df[k3][j]<<'␣';
1つのd.f.を出力
atdf<<"␣␣␣␣";
k個のd.f.を出力
atdf<<endl<<endl<<endl;
atdfをクローズ
end

# 最悪計算時間の見積り

a. 計算主要部の最悪計算時間を見積る。『$M \leqq n < N$な異なる要素数$n$に対し』、『現在の異なる$k$個のd.f.から』、『$(n+1)$に対するすべてのd.f.を作る』『まず、第$j$番目の値に1を加える』の4つのループが主要部であるから、フローチャートより

$$(N-M+1)\{k \cdot M\{M+3+nk(M)+M\}+k \cdot M\}$$

$$=(N-M+1)\{k \cdot M\{(nk+2)M+3\}+k \cdot M\}$$

$$=(N-M+1)\{k \cdot nk \cdot M^2+2k \cdot M^2+4 \cdot k \cdot M\}$$

$$\approx (N-M+1)\{(kMAX)^2+2(kMAX)\}M^2$$

$$\approx (kMAX)^2 \cdot N \cdot M^2$$

# MNdf の机上デバッグ

MMAX＝22，NMAX＝100（NMAXの値はデータサイズに無関係），
kMAX＝300，M＝3，N＝7 の時の机上デバッグ（desk debug）
M-1＝2

| k | nk | n | kk | j | df[ ][ ] 0 1 2 3 4 5 6 7 | pdf[ ] 0 1 2 3 4 5 6 7 | ndf[ ][ ] 0 1 2 3 4 5 6 7 | k3 | 備考 m |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 3 | 1 | 0 | 1 1 1 [0] | 2 1 1 | 2 1 1 | | |
| | 1 | | | 1 | | 1 2 1 | | | |
| | | | | 2 | | 1 1 2 | | | |
| 1 | 0 | | 2 | 3 | | | | | |
| | 1 | 4 | 1 | 0 | 2 1 1 [0] | 3 1 1 | 3 1 1 [0] | | |
| | | | | 1 | | 2 2 1 | | 0 | 0 m≠M |
| | 2 | | | | | | 2 2 1 [1] | 1 | |
| | | | | 2 | | 2 1 2 | | | |
| 2 | 0 | | 2 | 3 | | | | | |
| | | | | | 3 1 1 [0] | | | | |
| | 1 | 5 | 1 | 0 | 2 2 1 [1] | 4 1 1 | 4 1 1 [0] | | |
| | | | | 1 | | 3 2 1 | | 0 | 0 m≠M |
| | 2 | | | | | | 3 2 1 [1] | 1 | |
| | | | | 2 | | 3 1 2 | | | |
| | | | 2 | 3 | | | | | |
| | | | | 0 | | 3 2 1 | | 0 | 0 m≠M |
| | | | | | | | | 1 | 0 |
| | | | | | | | | | 1 |
| | | | | | | | | | 2 |
| | | | | | | | | | 3 |
| | | | | 1 | | 2 3 1 | | | |
| | | | | 2 | | 2 2 2 | | 0 | 0 m≠M |
| | | | | | | | | 1 | 0 m≠M |
| | 3 | | 3 | 3 | | | 2 2 2 [2] | 2 | |
| 3 | 0 | | | | | | | | |
| | | | | | 4 1 1 [0] | | | | |
| | | | | | 3 2 1 [1] | | | | |
| | | 6 | | | 2 2 2 [2] | | | | |
| | 1 | | 1 | 0 | | 5 1 1 | 5 1 1 [0] | | |
| | | | | 1 | | 4 2 1 | | 0 | 0 m≠M |
| | 2 | | | 2 | | 4 1 2 | 4 2 1 [1] | 1 | |
| | | | 2 | 3 | | 4 2 1 | | | |
| | | | | 0 | | | | 0 | 0 m≠M |
| | | | | | | | | 1 | 0 |
| | | | | | | | | | 1 |
| | | | | | | | | | 2 |
| | | | | | | | | | 3 m=M |
| | | | | 1 | | 3 3 1 | | 0 | 0 m≠M |
| | | | | | | | | 1 | 0 m≠M |
| | 3 | | | 2 | | 3 2 2 | 3 3 1 [2] | 2 | |

# MNdf の 机上デバッグ

MMAX＝22, NMAX＝100 (NMAXの値はデータサイズに無関係),

kMAX＝300, M=3, N=7 の時の机上デバッグ (desk debug)

| k | nk | n | kk | j | df[ ][ ] 0 1 2 3 4 5 6 7 | pdf[ ] 0 1 2 3 4 5 6 7 | ndf[ ][ ] 0 1 2 3 4 5 6 7 | k3 | 備考 m |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 0 | 0 m≠M |
| | | | | | | | | 1 | 0 m≠M |
| | | | | | | | | 2 | 0 |
| | 4 | | 3 | 3 | | | 3 2 2 [3] | 3 | 1 m≠M |
| | | | | 0 | | 3 2 2 | | 0 | 0 m≠M |
| | | | | | | | | 1 | 0 m≠M |
| | | | | | | | | 2 | 0 |
| | | | | | | | | | 7 $\overset{2\neq3}{m\neq M}$ |
| | | | | | | | | 3 | 0 |
| | | | | | | | | | 1 |
| | | | | | | | | | 2 |
| | | | | 1 | | 2 3 2 | | | 3 m=M |
| | | | | 2 | | 2 2 3 | | | |
| 4 | 0 | | 4 | 3 | | | | | |
| | | | | | 5 1 1 [0] | | | 0 | |
| | | | | | 4 2 1 [1] | | | 1 | |
| | | | | | 3 3 1 [2] | | | 2 | |
| | | 7 | | | 3 2 2 [3] | | | 3 | |

No.␣of␣distribution␣forms␣for␣n=M=␣7␣is␣4, where␣M␣=␣3,␣

N␣=␣7↵

↵

5␣1␣1␣␣␣␣␣4␣2␣1␣␣␣␣␣3␣3␣1␣␣␣␣␣3␣2␣2␣␣␣␣

↵

↵

↵

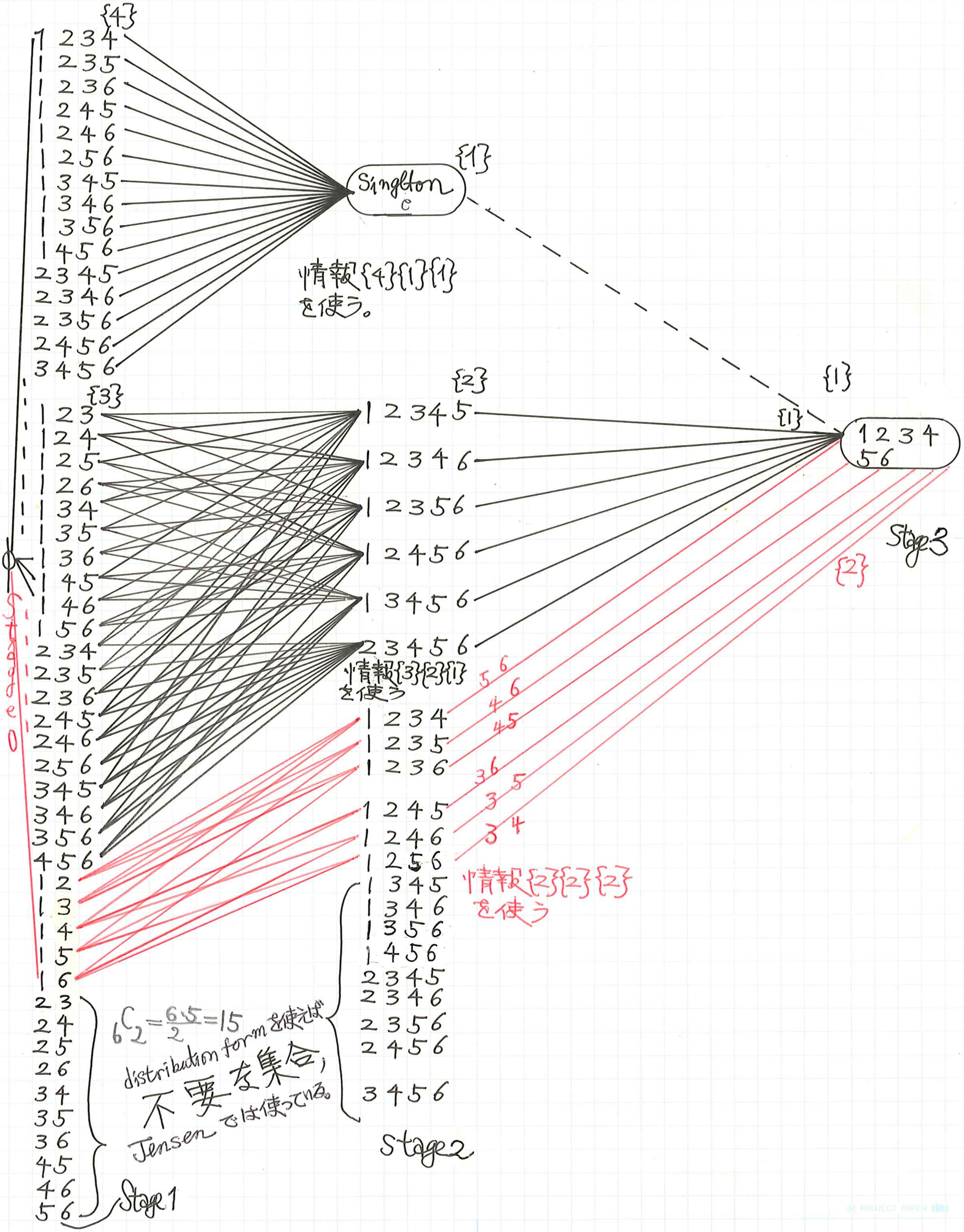
↵

All Right! by K. Iwamura 2003.12.13.

# M=3, N=6の時のdistribution formsとnetwork, $M_0=3$ as $N=6\geq 6=2M$ (Singlton$_c$ 利用)



・3つのdistribution forms {4}{1}{1}, {3}{2}{1}, {2}{2}{2}を持つ。



# M=4, N=8の時の distribution form {2}{2}{2}{2} に対する全partition 2520

${}_4C_2=\frac{4\cdot3}{2}=6$, ${}_2C_2=1$. $={}_8C_2\,{}_6C_2\,{}_4C_2\,{}_2C_2=\frac{8\cdot7}{2}\cdot\frac{6\cdot5}{2}\cdot\frac{4\cdot3}{2}=28\cdot15\cdot6=28\cdot90$

| | | | | |
|---|---|---|---|---|
| 12 | 34 | 56 | 78 | |
| 12 | 34 | 57 | 68 | |
| 12 | 34 | 58 | 67 | |
| ~~12~~ | ~~34~~ | ~~67~~ | ~~58~~ | |
| ~~12~~ | ~~34~~ | ~~68~~ | ~~57~~ | |
| ~~12~~ | ~~34~~ | ~~78~~ | ~~56~~ | |
| 12 | 35 | 46 | 78 | |
| 12 | 35 | 47 | 68 | |
| 12 | 35 | 48 | 67 | α |
| | | ~~67~~ | ~~48~~ | |
| | | ~~68~~ | ~~47~~ | |
| | | ~~78~~ | ~~46~~ | |
| 12 | 36 | 45 | 78 | |
| 12 | 36 | 47 | 58 | |
| 12 | 36 | 48 | 57 | |
| | | ~~57~~ | ~~48~~ | |
| | | ~~58~~ | ~~47~~ | |
| | | ~~78~~ | ~~45~~ | |
| 12 | 37 | 45 | 68 | |
| 12 | 37 | 46 | 58 | |
| 12 | 37 | 48 | 56 | |
| | | ~~56~~ | ~~48~~ | |
| | | ~~58~~ | ~~46~~ | |
| | | ~~68~~ | ~~45~~ | |
| 12 | 38 | 45 | 67 | |
| 12 | 38 | 46 | 57 | |
| 12 | 38 | 47 | 56 | 15 |
| | | ~~56~~ | ~~47~~ | |
| | | ~~57~~ | ~~46~~ | |
| | | ~~67~~ | ~~45~~ | |
| 12 | 45 | 36 | 78 | • |
| 12 | 45 | 37 | 68 | • |
| 12 | 45 | 38 | 67 | • |
| | | ~~67~~ | ~~38~~ | |
| | | ~~68~~ | ~~37~~ | |
| | | ~~78~~ | ~~36~~ | |
| 12 | 46 | 35 | 78 | • |
| 12 | 46 | 37 | 58 | • |
| 12 | 46 | 38 | 57 | • |
| | | ~~57~~ | ~~38~~ | |
| | | ~~58~~ | ~~37~~ | |
| | | ~~78~~ | ~~35~~ | |
| 12 | 47 | 35 | 68 | • |
| 12 | 47 | 36 | 58 | • |
| 12 | 47 | 38 | 56 | • |
| 12 | 48 | 35 | 67 | • |
| 12 | 48 | 36 | 57 | • |
| 12 | 48 | 37 | 56 | • |

| | | | | |
|---|---|---|---|---|
| 12 | 56 | 34 | 78 | • |
| 12 | 56 | 37 | 48 | • |
| 12 | 56 | 38 | 47 | • |
| 12 | 57 | 34 | 68 | • |
| 12 | 57 | 36 | 48 | • |
| 12 | 57 | 38 | 46 | • |
| 12 | 58 | 34 | 67 | • |
| 12 | 58 | 36 | 47 | • |
| 12 | 58 | 37 | 46 | • |
| 12 | 67 | 34 | 58 | • |
| 12 | 67 | 35 | 48 | • |
| 12 | 67 | 38 | 45 | • |
| 12 | 68 | 34 | 57 | • |
| 12 | 68 | 35 | 47 | • |
| 12 | 68 | 37 | 45 | • |
| 12 | 78 | 34 | 56 | • |
| 12 | 78 | 35 | 46 | • |
| 12 | 78 | 36 | 45 | • |
| 13 | 24 | 56 | 78 | |
| 13 | 24 | 57 | 68 | |
| 13 | 24 | 58 | 67 | |
| 13 | 25 | 46 | 78 | |
| 13 | 25 | 47 | 68 | |
| 13 | 25 | 48 | 67 | |
| 13 | 26 | 45 | 78 | |
| 13 | 26 | 47 | 58 | |
| 13 | 26 | 48 | 57 | |
| 13 | 27 | 45 | 68 | |
| 13 | 27 | 46 | 58 | |
| 13 | 27 | 48 | 56 | |
| 13 | 28 | 45 | 67 | |
| 13 | 28 | 46 | 57 | |
| 13 | 28 | 47 | 56 | |
| 13 | 45 | 26 | 78 | • |
| 13 | 45 | 27 | 68 | • |
| 13 | 45 | 28 | 67 | • |
| 13 | 46 | 25 | 78 | o |
| 13 | 47 | 25 | 68 | o |
| 13 | 48 | 25 | 67 | o |

| | | | | |
|---|---|---|---|---|
| 13 | 56 | 24 | 78 | o |
| 13 | 57 | 24 | 68 | o |
| 13 | 58 | 24 | 67 | o |
| 13 | 67 | 24 | 58 | o |
| 13 | 68 | 24 | 57 | o |
| 13 | 78 | 24 | 56 | o |
| 14 | 23 | 56 | 78 | |
| 14 | 23 | 57 | 68 | |
| 14 | 23 | 58 | 67 | |
| 14 | 25 | 36 | 78 | |
| 14 | 25 | 37 | 68 | |
| 14 | 25 | 38 | 67 | |
| 14 | 26 | 35 | 78 | |
| 14 | 26 | 37 | 58 | |
| 14 | 26 | 38 | 57 | |
| 14 | 27 | 35 | 68 | |
| | | 36 | 58 | |
| | | 38 | 56 | |
| 14 | 28 | 35 | 67 | |
| | | 36 | 57 | |
| | | 37 | 56 | |
| 14 | 35 | 26 | 78 | o |
| 14 | 36 | 25 | 78 | o |
| 14 | 37 | 25 | 68 | o |
| 14 | 38 | 25 | 67 | o |
| 14 | 56 | 23 | 78 | o |
| 14 | 57 | 23 | 68 | o |
| 14 | 58 | 23 | 67 | o |
| 14 | 67 | 23 | 58 | o |
| 14 | 68 | 23 | 57 | o |
| 14 | 78 | 23 | 56 | o |
| 15 | 23 | 46 | 78 | |
| | | 47 | 68 | |
| | | 48 | 67 | α |
| 15 | 24 | 36 | 78 | |
| | | 37 | 68 | |
| | | 38 | 67 | |
| 15 | 26 | 34 | 78 | |
| | | 37 | 47 | |
| | | 38 | 47 | |

| | | | | |
|---|---|---|---|---|
| 15 | 27 | 34 | 68 | |
| | | 36 | 48 | |
| | | 38 | 46 | |
| 15 | 28 | 34 | 67 | |
| | | 36 | 47 | |
| | | 37 | 46 | |
| 15 | 34 | 26 | 78 | o |
| 15 | 36 | 24 | 78 | o |
| 15 | 37 | 24 | 68 | o |
| 15 | 38 | 24 | 67 | o |
| 15 | 46 | 23 | 78 | o |
| 15 | 47 | 23 | 68 | o |
| 15 | 48 | 23 | 67 | o |
| 15 | 67 | 23 | 48 | o |
| 15 | 68 | 23 | 47 | o |
| 15 | 78 | 23 | 46 | o |
| 16 | 23 | 45 | 78 | |
| | | 47 | 58 | |
| | | 48 | 57 | |
| 16 | 24 | 35 | 78 | |
| | | 37 | 58 | |
| | | 38 | 57 | |
| 16 | 25 | 34 | 78 | |
| | | 37 | 48 | |
| | | 38 | 47 | |
| 16 | 27 | 34 | 58 | |
| | | 35 | 48 | |
| | | 38 | 45 | |
| 16 | 28 | 34 | 57 | |
| | | 35 | 47 | |
| | | 37 | 45 | |
| 16 | 34 | 25 | 78 | o |
| 16 | 35 | 24 | 78 | o |
| 16 | 37 | 24 | 58 | o |
| 16 | 38 | 24 | 57 | o |
| 16 | 45 | 23 | 78 | o |
| 16 | 47 | 23 | 58 | o |
| 16 | 48 | 23 | 57 | o |
| 16 | 57 | 23 | 48 | o |
| 16 | 58 | 23 | 47 | o |
| 16 | 78 | 23 | 45 | o |

「•」,「o」は duplicate partition であることを示す。oは下2つが異なる3つのpartitionをまとめた。

# M=4, N=8の時のdistribution form{2}{2}{2}{2}に対する全partition

| | | | | |
|---|---|---|---|---|
| 17 | 23 | 45 | 68 | |
| | | 46 | 58 | |
| | | 48 | 56 | |
| 17 | 24 | 35 | 68 | |
| | | 36 | 58 | |
| | | 38 | 56 | |
| 17 | 25 | 34 | 68 | |
| | | 36 | 48 | |
| | | 38 | 46 | |
| 17 | 26 | 34 | 58 | |
| | | 35 | 48 | |
| | | 38 | 45 | |
| 17 | 28 | 34 | 56 | |
| | | 35 | 46 | |
| | | 36 | 45 | |
| 17 | 34 | 25 | 68 | ○ |
| 17 | 35 | 24 | 68 | ○ |
| 17 | 36 | 24 | 58 | ○ |
| 17 | 38 | 24 | 56 | ○ |
| 17 | 45 | 23 | 68 | ○ |
| 17 | 46 | 23 | 58 | ○ |
| 17 | 48 | 23 | 56 | ○ |
| 17 | 56 | 23 | 48 | ○ |
| 17 | 58 | 23 | 46 | ○ |
| 17 | 68 | 23 | 45 | ○ |
| 18 | 23 | 45 | 67 | |
| | | 46 | 57 | |
| | | 47 | 56 | |
| 18 | 24 | 35 | 67 | |
| | | 36 | 57 | |
| | | 37 | 56 | |
| 18 | 25 | 34 | 67 | |
| | | 36 | 47 | |
| | | 37 | 46 | |
| 18 | 26 | 34 | 57 | |
| | | 35 | 47 | |
| | | 37 | 45 | |
| 18 | 27 | 34 | 56 | |
| | | 35 | 46 | |
| | | 36 | 45 | |

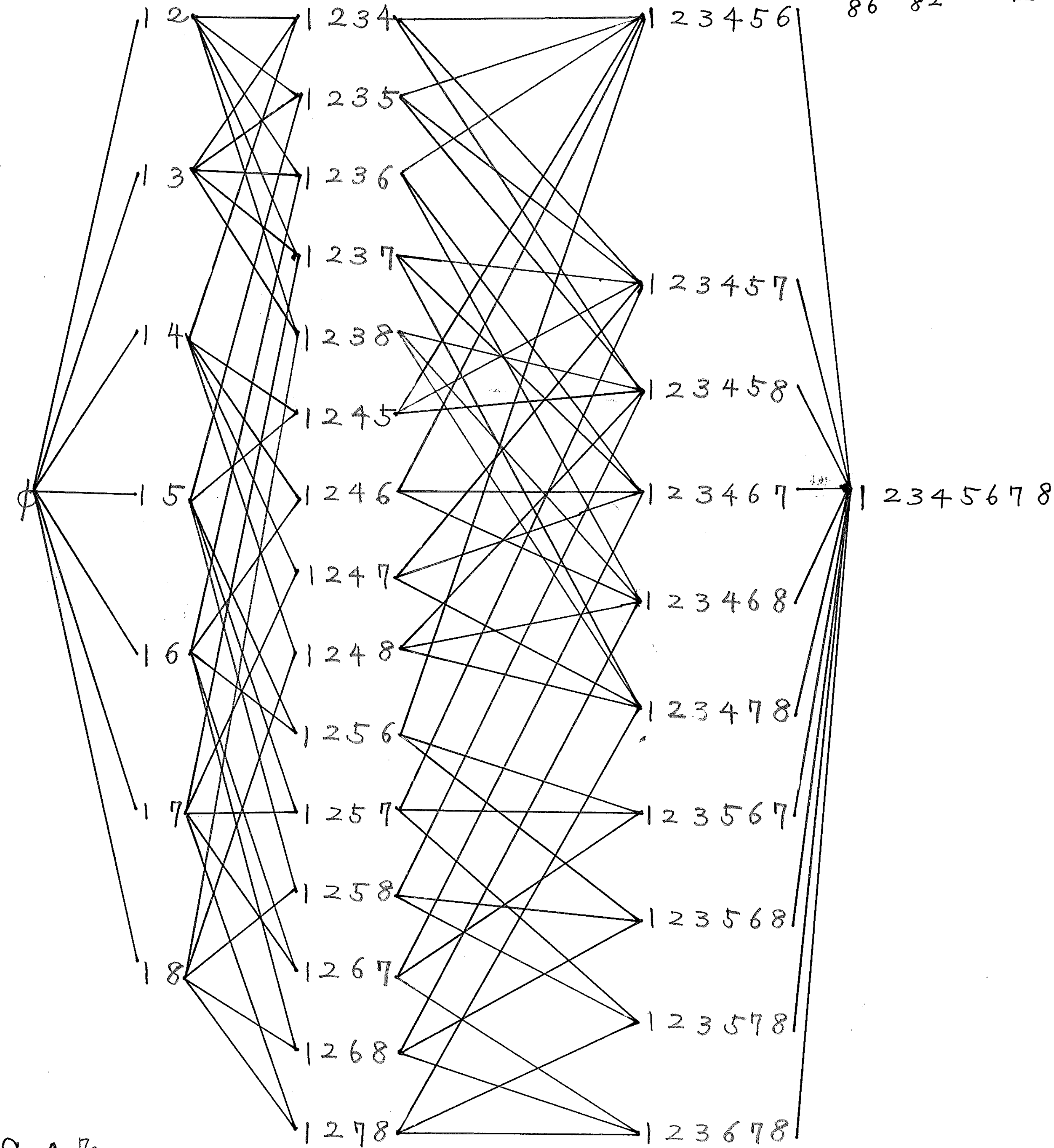
| | | | | |
|---|---|---|---|---|
| 18 | 34 | 25 | 67 | ○ |
| 18 | 35 | 24 | 67 | ○ |
| 18 | 36 | 24 | 57 | ○ |
| 18 | 37 | 24 | 56 | ○ |
| 18 | 45 | 23 | 67 | ○ |
| 18 | 46 | 23 | 57 | ○ |
| 18 | 47 | 23 | 56 | ○ |
| 18 | 56 | 23 | 47 | ○ |
| 18 | 57 | 23 | 46 | ○ |
| 18 | 67 | 23 | 45 | ○ |
| ~~18~~ | ~~78~~ | | | |
| 23 | 14 | 56 | 78 | ✓ |
| 23 | 15 | 46 | 78 | ✓ |

2003.12.15-IV

M=4, N=8の時の distribution form {2}{2}{2}{2} に対する全partitionから作られる network (26)

${}_8C_2=\frac{8\cdot 7}{2}=28$, ${}_6C_2=\frac{6\cdot 5}{2}=15$, ${}_4C_2=\frac{4\cdot 3}{2}=6$, ${}_2C_2=1$, ${}_8C_4=\frac{8\cdot 7\cdot 6\cdot 5}{4\cdot 3\cdot 2}=7\cdot 2\cdot 5=70$,

${}_8C_6={}_8C_2=28$ □




Stage0 7arcs

35arcs

15×3=45arcs

10arcs

Stage1 7states

Stage2 15states

Stage3 10states

Stage4

7+35+45+10=97arcs in all, in 7×15=105 partitions

下手にnetworkを作ると 105×2+105×3=105×5=525 arcs 必要となる。

No. of distribution forms for n=M= 24 is 199, where M = 6 N = 24

N

```
k3/100 = 0

19 1 1 1 1 1    18 2 1 1 1 1    17 3 1 1 1 1    17 2 2 1 1 1    16 4 1 1 1 1    16 3 2 1 1 1    16
2 2 2 1 1    15 5 1 1 1 1    15 4 2 1 1 1    15 3 3 1 1 1
15 3 2 2 1 1    15 2 2 2 2 1    14 6 1 1 1 1    14 5 2 1 1 1    14 4 3 1 1 1    14 4 2 2 1 1    14
3 3 2 1 1    14 3 2 2 2 1    14 2 2 2 2 2    13 7 1 1 1 1
13 6 2 1 1 1    13 5 3 1 1 1    13 5 2 2 1 1    13 4 4 1 1 1    13 4 3 2 1 1    13 4 2 2 2 1    13
3 3 3 1 1    13 3 3 2 2 1    13 3 2 2 2 2    12 8 1 1 1 1
12 7 2 1 1 1    12 6 3 1 1 1    12 6 2 2 1 1    12 5 4 1 1 1    12 5 3 2 1 1    12 5 2 2 2 1    12
4 4 2 1 1    12 4 3 3 1 1    12 4 3 2 2 1    12 4 2 2 2 2
12 3 3 3 2 1    12 3 3 2 2 2    11 9 1 1 1 1    11 8 2 1 1 1    11 7 3 1 1 1    11 7 2 2 1 1    11
6 4 1 1 1    11 6 3 2 1 1    11 6 2 2 2 1    11 5 5 1 1 1
11 5 4 2 1 1    11 5 3 3 1 1    11 5 3 2 2 1    11 5 2 2 2 2    11 4 4 3 1 1    11 4 4 2 2 1    11
4 3 3 2 1    11 4 3 2 2 2    11 3 3 3 3 1    11 3 3 3 2 2
10 10 1 1 1 1    10 9 2 1 1 1    10 8 3 1 1 1    10 8 2 2 1 1    10 7 4 1 1 1    10 7 3 2 1 1    10
7 2 2 2 1    10 6 5 1 1 1    10 6 4 2 1 1    10 6 3 3 1 1
10 6 3 2 2 1    10 6 2 2 2 2    10 5 5 2 1 1    10 5 4 3 1 1    10 5 4 2 2 1    10 5 3 3 2 1    10
5 3 2 2 2    10 4 4 4 1 1    10 4 4 3 2 1    10 4 4 2 2 2
10 4 3 3 3 1    10 4 3 3 2 2    10 3 3 3 3 2    9 9 3 1 1 1    9 9 2 2 1 1    9 8 4 1 1 1    9 8 3
2 1 1    9 8 2 2 2 1    9 7 5 1 1 1    9 7 4 2 1 1
9 7 3 3 1 1    9 7 3 2 2 1    9 7 2 2 2 2    9 6 6 1 1 1    9 6 5 2 1 1    9 6 4 3 1 1    9 6 4 2 2
1    9 6 3 3 2 1    9 6 3 2 2 2    9 5 5 3 1 1
k3/100 = 1

9 5 5 2 2 1    9 5 4 4 1 1    9 5 4 3 2 1    9 5 4 2 2 2    9 5 3 3 3 1    9 5 3 3 2 2    9 4 4 4 2
1    9 4 4 3 3 1    9 4 4 3 2 2    9 4 3 3 3 2
9 3 3 3 3 3    8 8 5 1 1 1    8 8 4 2 1 1    8 8 3 3 1 1    8 8 3 2 2 1    8 8 2 2 2 2    8 7 6 1 1
1    8 7 5 2 1 1    8 7 4 3 1 1    8 7 4 2 2 1
8 7 3 3 2 1    8 7 3 2 2 2    8 6 6 2 1 1    8 6 5 3 1 1    8 6 5 2 2 1    8 6 4 4 1 1    8 6 4 3 2
1    8 6 4 2 2 2    8 6 3 3 3 1    8 6 3 3 2 2
8 5 5 4 1 1    8 5 5 3 2 1    8 5 5 2 2 2    8 5 4 4 2 1    8 5 4 3 3 1    8 5 4 3 2 2    8 5 3 3 3
2    8 4 4 4 3 1    8 4 4 4 2 2    8 4 4 3 3 2
8 4 3 3 3 3    7 7 7 1 1 1    7 7 6 2 1 1    7 7 5 3 1 1    7 7 5 2 2 1    7 7 4 4 1 1    7 7 4 3 2
1    7 7 4 2 2 2    7 7 3 3 3 1    7 7 3 3 2 2
7 6 6 3 1 1    7 6 6 2 2 1    7 6 5 4 1 1    7 6 5 3 2 1    7 6 5 2 2 2    7 6 4 4 2 1    7 6 4 3 3
1    7 6 4 3 2 2    7 6 3 3 3 2    7 5 5 5 1 1
7 5 5 4 2 1    7 5 5 3 3 1    7 5 5 3 2 2    7 5 4 4 3 1    7 5 4 4 2 2    7 5 4 3 3 2    7 5 3 3 3
3    7 4 4 4 4 1    7 4 4 4 3 2    7 4 4 3 3 3
6 6 6 4 1 1    6 6 6 3 2 1    6 6 6 2 2 2    6 6 5 5 1 1    6 6 5 4 2 1    6 6 5 3 3 1    6 6 5 3 2
2    6 6 4 4 3 1    6 6 4 4 2 2    6 6 4 3 3 2
6 6 3 3 3 3    6 5 5 5 2 1    6 5 5 4 3 1    6 5 5 4 2 2    6 5 5 3 3 2    6 5 4 4 4 1    6 5 4 4 3
2    6 5 4 3 3 3    6 4 4 4 4 2    6 4 4 4 3 3
5 5 5 5 3 1    5 5 5 5 2 2    5 5 5 4 4 1    5 5 5 4 3 2    5 5 5 3 3 3    5 5 4 4 4 2    5 5 4 4 3
3    5 4 4 4 4 3    4 4 4 4 4 4
```

かかった時間は0時間0分3秒です。

# R.E.Jensen (Oper. Res. 17 (1969) 1034–1057)の論文で不明な点、

# 改善すべき点について

1. 2003.12.5-V, VII で示したように異なった distribution form 間での path の移動を認めると、非効率が生じるケースがある。2003.12.5-VII で Stage 1 の States 1234 や 123 が Stage 2 の state 1234 の真部分集合でない ($\{1,2,3,4\}=\{1,2,3,4\}$)、$\{1,2\}$ から $\{1,2,3,4,5\}$ への path は無いのに $\{1,2\}\subsetneqq\{1,2,3,4,5\}$, 12 から 12345 に arc があると誤認、などの無駄が生じてしまう。各 distribution form 毎に DP を適用する場合はこの様な無駄が生じない。もっとも、Jensen は redundancy arc の除去でこの様な不都合を避けると主張している。

2. Jensen 流でやると Stage 1 の state 総数 NS(1) がその他の Stage の State 総数以上であるので全 State 総数は $NS(1)*(M_0-1)$ で押えられる。しかし, Stage K, (K+1) 間に arc があるかどうかの check は

$$NS(K) * NS(K+1) \text{ ---- (a)}$$

回しなければならない。この回数は各 distribution form 毎に DP 適用した時の number of states $ns(K, kk)$ を用いると

$$ns(K, kk) * ns(K+1, kk) \text{ --- (b)}$$

でこれは (a) よりぐっと小さい。もちろん $\sum_{kk=1}^{k}\sum_{K=1}^{M_0-1} ns(K, kk)*ns(K+1, kk) < \sum_{K=1}^{M_0-1} NS(K)*NS(K+1)$ である。

3. "$M=3$, $N=6$, $M_0=3$ の distribution forms と network (Singleton 利用)" にある様に $\{4\}\{1\}\{1\}$ などの $\{1\}$ が複数個続く distribution form に対しては Singleton を利用したほうが arc があるかどうかの無駄な check を大幅に減らせる。

4. "M=4, N=8の時のdistribution form $\{2\}\{2\}\{2\}\{2\}$ に対する全partition",
"M=4, N=8の時のdistribution form $\{2\}\{2\}\{2\}\{2\}$ に対する全partitionから作られるnetwork"が示すようにStage 1からStage 3までのstate数, arc数は以下の様になる。このdistribution formにJensenの考え方を適用した場合と最小state数最小arc数との比較:

$_8C_2=\frac{8\cdot 7}{2}=28$, $_8C_4=\frac{8\cdot 7\cdot 6\cdot 5}{4\cdot 3\cdot 2}=10\cdot 7=70$, $_8C_6={}_8C_2=28$

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 最小 | 1 state | 7 arcs | 7 states | 35 arcs | 15 states | 45 arcs | 10 states | 10 arcs | 1 state | 34 states 97 arcs |
| Jensen | 1 state | 28 arcs | 28 states | 1960 arcs | 70 states | 1960 arcs | 28 states | 28 arcs | 1 state | 128 states 3976 arcs |
| | Stage 0 | | Stage 1 | | Stage 2 | | Stage 3 | | Stage 4 | |

この表より最小state数最小arc数の構成の方が一般的には優れているが、これを作成する為にはかなりの余分な計算時間が必要となる。Fuzzy Clusteringの場合にはnetworkを100回, 1000回と使うので, Jensenの方法との比較は工学上の問題となる。実用プログラミング学生実習課題として将来計算実験する価値はある。

5. 各Stage Kでのstatesの生成は難かしくはない。Jensenの方法でも最小state数最小arc数法でもstatesの生成・表現は簡単である。

6. $2\leqq M\leqq N\leqq 32$ を仮定する。どのStageにおいてもstateは全て番号付けしておく(既に上から下に一列に並べられているとする)。Stage Kのstate sk とStage (K+1)のstate skp1 とにおいて

sk が skp1 の部分集合かどうかは"ビット毎の &"に対し

sk==(sk&skp1)

を check すればよい。yes ならば部分集合であり no ならば部分集合ではない。部分集合の時、sk と skp1 の exclusive or を取って その結果 all bits 0 なら集合として等しく, all bits 0 でなければ真部分集合である。

$2 \leqq M \leqq N \leqq 32$ の仮定から $1 \leqq i \leqq 32$ なる $i$ を state $S$ が含むのは $S$ の第 $(i-1)$ ビット(または $S$ の第 $(32-i)$ ビット)目が 1 のときと定義すればよい。第 $(i-1)$ ビット目を使うか、第 $(32-i)$ ビット目を使うかは後日決定する。

7. $N=24$, $M=6$ に対する 199 個の distribution forms は MNdf.cpp によって 3 秒 で全て計算出来た(TECRA 9000, 1.2GHz Pentium IV, 1GB メモリ)。必要メモリはサイズは約 54KB。従って distribution forms を使って悪いということはない。

8. $M=3$, $N=6$, $M_0=3$ の時の Jensen の方法による Stage 毎の state 数:

$$NS(1)=\sum_{L=2}^{4}\binom{6}{L}=\binom{6}{2}+\binom{6}{3}+\binom{6}{4}=\frac{6\cdot 5}{2}+\frac{6\cdot 5\cdot 4}{3\cdot 2}+3\cdot 5=15+20+15$$
$$=50,$$
$$NS(2)=\sum_{L=4}^{5}\binom{6}{L}=\binom{6}{4}+\binom{6}{5}=\binom{6}{2}+\binom{6}{1}=15+6=21,$$
$$NS(3)=\sum_{L=6}^{6}\binom{6}{L}=\binom{6}{6}=1.$$

3 個の distribution forms 毎に DP を適用する時は $ns(K, k)$ を用いて

$$ns(1,1)=\binom{6}{4}=15,\quad ns(1,2)=\binom{6}{3}=20,\quad ns(1,3)=\binom{6}{2}=15,$$
$$ns(2,1)=\binom{6}{5}=6,\quad ns(2,2)=\binom{6}{5}=6,\quad ns(2,3)=\binom{6}{4}=15,$$
$$ns(3,1)=\binom{6}{6}=1,\quad ns(3,2)=\binom{6}{6}=1,\quad ns(3,3)=\binom{6}{6}=1.$$

8.(続き)

$$\sum_{kk=1}^{3}\sum_{K=1}^{2} ns(K,kk)*ns(K+1,kk)$$

$= ns(1,1)*ns(2,1)+ns(2,1)*ns(3,1)$

$+ns(1,2)*ns(2,2)+ns(2,2)*ns(3,2)$

$+ns(1,3)*ns(2,3)+ns(2,3)*ns(3,3)$

$$\begin{array}{r} 96 \\ 126 \\ +)\ 240 \\ \hline 462 \end{array}$$

$=(15\cdot 6+6\cdot 1)+(20\cdot 6+6\cdot 1)+(15\cdot 15+15\cdot 1)=96+126+240=462$ //。

さて、Jensenの方法では

$$\sum_{K=1}^{2} NS(K)*NS(K+1)=50\cdot 21+21\cdot 1=1250+21=1271 \text{ //。}$$

9. $M=3$, $N=7>6=2M$, $M_0=3$の時のStage毎のstate数：

$\{5\}\{1\}\{1\}$

$\{4\}\{2\}\{1\}$

$\{3\}\{3\}\{1\}$

K $\{3\}\{2\}\{2\}$

$NS(1)=\binom{7}{3}+\binom{7}{4}+\binom{7}{5}=91$,

$NS(2)=\binom{7}{5}+\binom{7}{6}=\binom{7}{2}+\binom{7}{1}=\frac{7\cdot 6}{2}+7=28$,

$NS(3)=1$.

$ns(1,\underline{1})=\binom{7}{5}=21$, $ns(1,\underline{2})=\binom{7}{4}=35$, $ns(1,\underline{3})=\binom{7}{3}=35$, $ns(1,\underline{4})=\binom{7}{3}=35$,

$ns(2,1)=\binom{7}{6}=7$, $ns(2,2)=\binom{7}{6}=7$, $ns(3,3)=\binom{7}{6}=7$, $ns(2,4)=\binom{7}{5}=21$,

$ns(3,1)=\binom{7}{7}=1$, $ns(3,2)=\binom{7}{7}=1$, $ns(3,3)=\binom{7}{7}=1$, $ns(3,4)=\binom{7}{7}=1$.

$$\sum_{kk=1}^{4}\sum_{K=1}^{2} ns(K,kk)*ns(K+1,kk)=(21\cdot 7+7\cdot 1)+(35\cdot 7+7\cdot 1)+(35\cdot 7+7\cdot 1)$$

$+(35\cdot 21+21\cdot 1)=22\cdot 7+36\cdot 7+36\cdot 7+36\cdot 21=58\cdot 7+36\cdot 28=1414$ //。

$$\begin{array}{r} 58 \\ \times)\ 7 \\ \hline 406 \end{array} \quad \begin{array}{r} 36 \\ \times)\ 28 \\ \hline 288 \\ 72\ \\ \hline 1008 \end{array} \quad \begin{array}{r} 1008 \\ +)\ 406 \\ \hline 1414 \end{array}$$

$$\sum_{K=1}^{2} NS(K)*NS(K+1)=91\cdot 28+28=92\cdot 28=2576 \text{ //。}$$

$$\begin{array}{r} 92 \\ \times)\ 28 \\ \hline 736 \\ 184\ \\ \hline 2576 \end{array} \qquad \begin{array}{r} 1414 \\ 105 \\ +)\ 21 \\ \hline 1540 \end{array} \qquad \begin{array}{r} 2576 \\ +)\ 91 \\ \hline 2667 \end{array}$$

distribution forms毎のcheckingの時の総check回数は

$21+35+35+35+1414=21+105+1414=1540$ //

Jensen法の総check回数は $91+2576=2667$ //

$$\sum_{K=1}^{3} NS(K)=120<172=\sum_{kk=1}^{4}\sum_{K=1}^{3} ns(K,kk) \ !$$

# Jensenの論文1042ページの意味内容

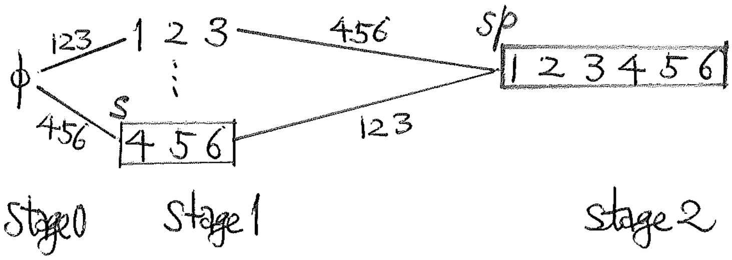


$M=3, N=7>6=2M, M_0=3$ で 全てのdistribution formsは以下の4つ。

{5}{1}{1}
{4}{2}{1}
{3}{3}{1}
{3}{2}{2}

| K= | 1 | 2 | 3 |
|---|---|---|---|
| max(K) | 5 | 6 | 7 |
| min(K) | 3 | 5 | 7 |

a. K=1での要素数3のstateとK=2での要素数6のstateとの間には

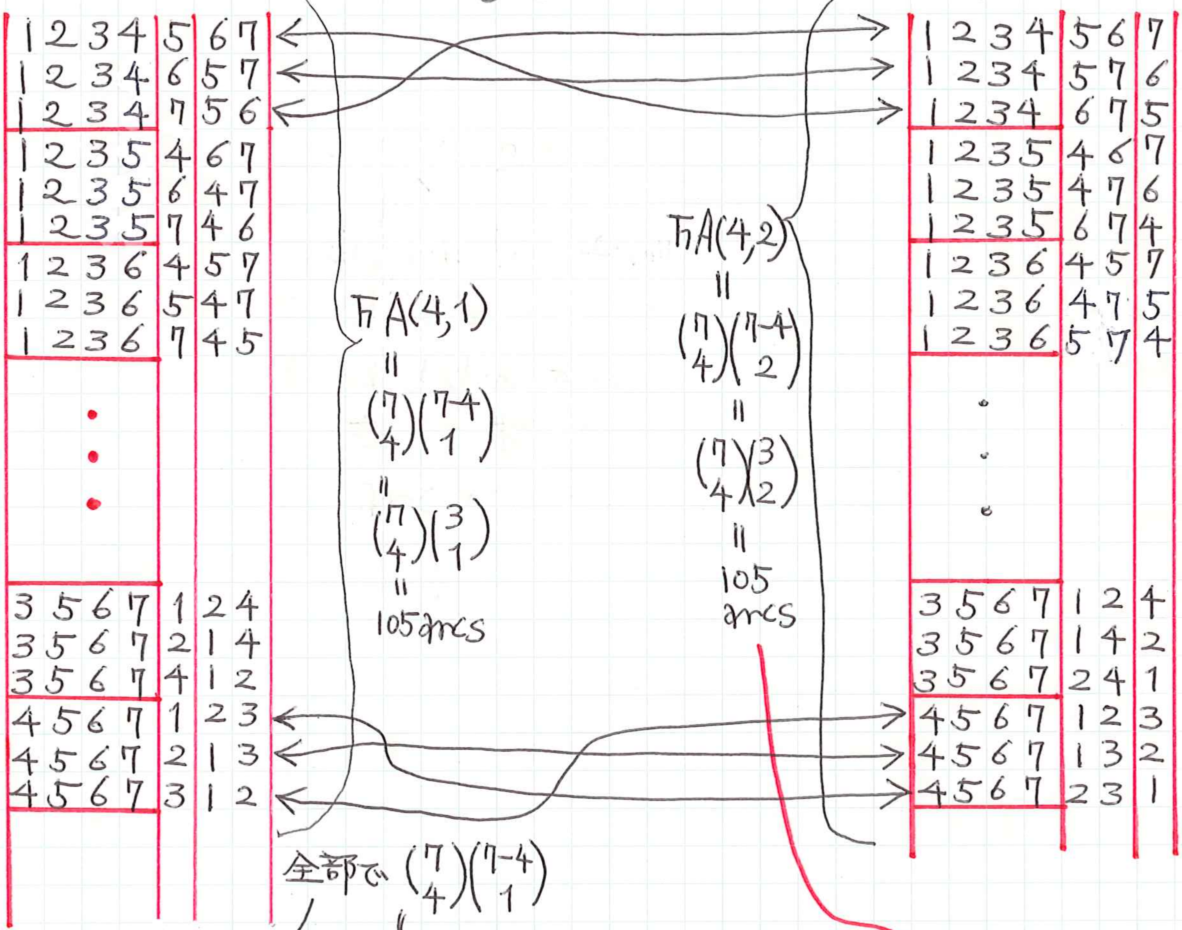


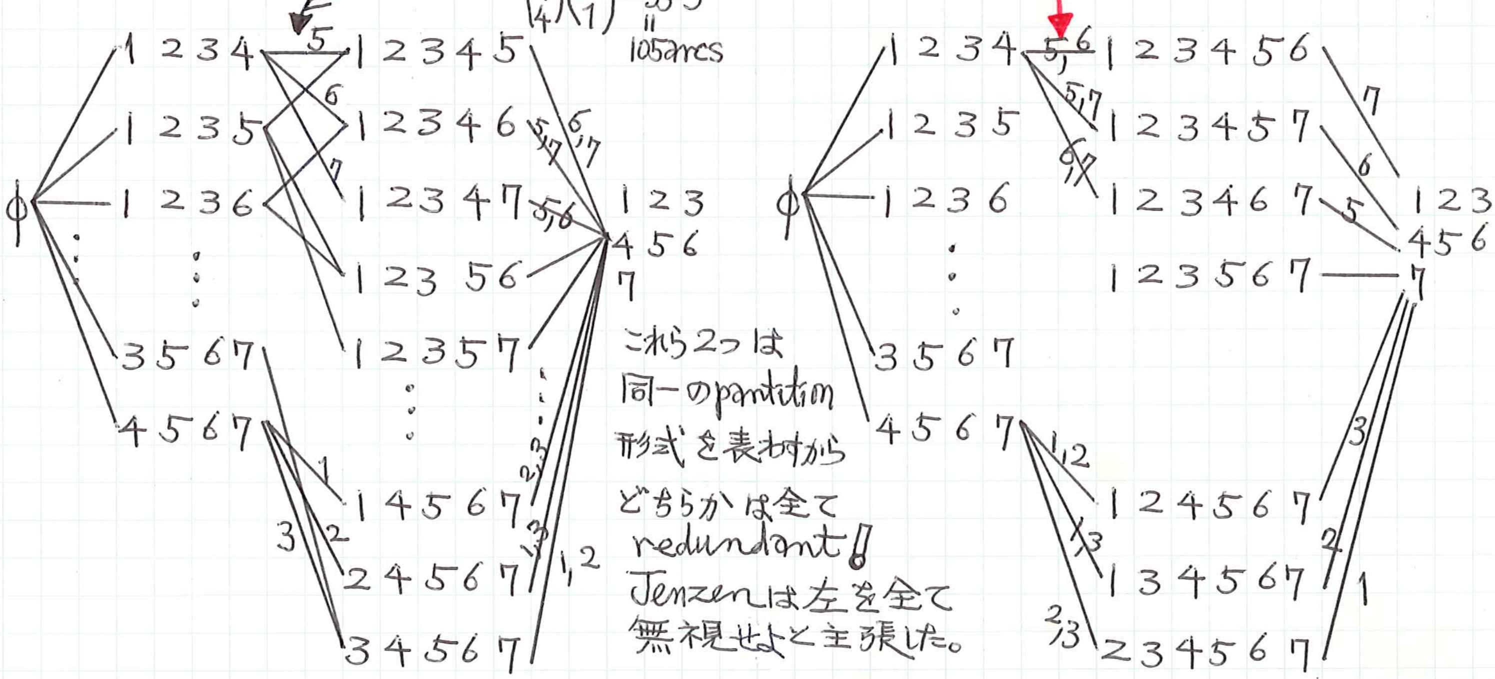

上図の様な関係が起る。K=1でのstate s とK=2でのstate sp を評価中とし、各stageにおいてarcがあるかどうか($s \subsetneq sp$のcheck)は上から下に順次行われるものとする。

(s & sp)==s でかつ [s (EXCLUSIVE OR) sp] ==r

となるstate rがK=1にあるならば(上述でr=123)このs=456とsp=123456に対するDP計算を実行しない。即ちForward DPに対しては arc (s, sp)はredundantであり、Backward DPに対しては arc(φ, s)はredundantであるので、これらのarcに対しては DP計算をしない。(s,sp)間のarcを物理的に除去はする必要がないことに注意せよ。各stateに対しarc接続情報を持たせることはしない。

b。Jensenの方法では 4元からなるstateがStage 1に $\binom{7}{4}$ 個あり、また4番目のdistribution formから $3+2=5$ 元からなるstateがStage 2にある。Jensenでは Stage 1のstateで stage 2のあるstateの部分集合(真部分)になるものがあれば、これらのstateにはarcがあると考えるので(!?)下記の様な状況が起こる。各distribution毎にarcsを考えれば このようなredundancyは起こらない。



これら2つは同一のpartition形式を表すから どちらかは全て redundant! Jenzenは左を全て無視せよと主張した。

$NS(2)=\sum_{L=4}^{18}\binom{32}{L}=NS(1)+\binom{32}{18}-\binom{32}{2}-\binom{32}{3}=NS(1)+\binom{32}{14}$ 正しい 32−18=14

$-\frac{32\cdot31}{2}-\frac{32\cdot31\cdot30}{2\cdot3}=NS(1)+\frac{32\cdot31\cdot30\cdot29\cdot28\cdot27\cdot26\cdot25\cdot24\cdot23\cdot22\cdot21}{14\cdot13\cdot12\cdot11\cdot10\cdot9\cdot8\cdot7\cdot6\cdot3\cdot2\cdot5\cdot4}$

$-16\cdot31-16\cdot31\cdot10=NS(1)+31\cdot29\cdot2\cdot3\cdot2\cdot25\cdot2\cdot23\cdot2\cdot19-16\cdot31\cdot11$

$=3013\,746530 \quad +471,435,600-5456$ 30億

$=3,485,176,674$ states $=3.49\times4=$ 13.96GB, $NS(1)=3,013,746,530$。

Memory 1〜2GBのパソコンでは statesをメモリ内に入れられない。パソコンでは処理不可能!! |int型データ|≦約21億

$N=32, M=16, M_0=16$のケースは不可能!!

---

$N=25, M=6$で $NS(1)=33523880 \fallingdotseq 134MB$, $N-M=19$

| K | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| max(K) | 20 | 21 | 22 | 23 | 24 | 25 |
| min(K) | 5 | 9 | 14 | 17 | 21 | 25 |

$N-M[N/M]=25-6\cdot4=1$, K=1に対し $((4+1)\cdot1=5)$.

K=2に対し $25-(6-2)\cdot4=25-16=9$, K=3に対し $26-(6-3)\cdot4=26-12=14$, K=4に対し $25-(6-4)\cdot4=25-8=17$, K=5に対し $25-(6-5)\cdot4=21$, K=6に対し $25-(6-6)\cdot4=25$

$\binom{25}{8}=\frac{25\cdot24\cdots19}{8\cdot\cdots\cdots}\cdot18=1,081,575$

$NS(1)=33523880$

$NS(2)=NS(1)+\binom{25}{21}-\binom{25}{5}-\binom{25}{6}-\binom{25}{7}-\binom{25}{8}$ $\binom{25}{4}$

$\binom{25}{4}=\frac{25\cdot24\cdot23\cdot22}{4\cdot3\cdot2}=12650$

$\binom{25}{5}=\frac{25\cdot24\cdot23\cdot22\cdot21}{5\cdot4\cdot3\cdot2}=\frac{12650}{5}\cdot21=2530\cdot21=53130$

$\binom{25}{6}=\frac{25\cdot24\cdot23\cdot22\cdot21\cdot20}{6\cdot5\cdot4\cdot3\cdot2}=\frac{20}{6}\cdot53130=8855\cdot20=177,100$

$\binom{25}{7}=\frac{25\cdot24\cdot23\cdot22\cdot21\cdot20\cdot19}{7\cdot6\cdot5\cdot4\cdot3\cdot2}=\frac{177,100}{7}\cdot19=25300\cdot19=480,700$

# M=6, N=25の時の 全stateの総数(続き)

$NS(2) = 33{,}523{,}880 + 12{,}650$
$\quad -(53{,}130 + 177{,}100 + 480{,}700 + 1{,}081{,}575)$
$\quad = 33{,}536{,}530 - 1{,}792{,}505 = 31{,}744{,}025$

$NS(3) = \sum_{L=14}^{22} \binom{25}{L} = NS(2) + \binom{25}{22} - \binom{25}{9} - \binom{25}{10} - \binom{25}{11} - \binom{25}{12} - \binom{25}{13}$

$\binom{25}{9} = \frac{(25\cdot\cdots 19\cdot 18)\cdot 17}{9(8\cdot 7\cdots 2)} = \binom{25}{8}\cdot\frac{17}{9} = 1081575\cdot\frac{17}{9} = 2{,}042{,}975$

$\binom{25}{10} = \frac{(25\cdot\cdots 17)\cdot 16}{10(9\cdots 2)} = \binom{25}{9}\cdot\frac{16}{10} = 3{,}268{,}760$

$\binom{25}{11} = \binom{25}{10}\cdot\frac{15}{11} = 3268760\cdot\frac{15}{11} = 4{,}457{,}400$

$\binom{25}{12} = \binom{25}{11}\cdot\frac{\cancel{14}\ 7}{\cancel{12}\ 6} = 4457400\cdot\frac{7}{6} = 5{,}200{,}300$

$\binom{25}{13} = 5{,}200{,}300$ +

⊖ $20{,}169{,}735$

$\binom{25}{22} = \binom{25}{3} = \frac{25\cdot 24\cdot 23}{3\cdot 2} = 2{,}300$, $\binom{25}{2} = \frac{25\cdot 24}{2} = 300$

$NS(3) = 11{,}576{,}590$

$NS(4) = \sum_{L=17}^{23}\binom{25}{L} = \binom{25}{17}+\binom{25}{18}+\binom{25}{19}+\binom{25}{20}+\binom{25}{21}+\binom{25}{22}+\binom{25}{23}$
$\quad = \binom{25}{8}+\binom{25}{7}+\binom{25}{6}+\binom{25}{5}+\binom{25}{4}+\binom{25}{3}+\binom{25}{2}$

$$
\begin{array}{r}
= 1081575 \\
480700 \\
177100 \\
53130 \\
12650 \\
2300 \\
300 \ + \\
\hline
= 1{,}807{,}755
\end{array}
$$

$NS(5) = \binom{25}{21}+\binom{25}{22}+\binom{25}{23}+\binom{25}{24} = \binom{25}{4}+\binom{25}{3}+\binom{25}{2}+\binom{25}{1} =$

$$
\begin{array}{r}
12{,}650 \\
2{,}300 \\
300 \\
25 \ + \\
\hline
= 15{,}275
\end{array}
$$

$NS(6) = 1.$

Stateの総数 = 33,523,800 = 78,667,446 ≈ $78.7\times 10^6$

$$
\begin{array}{r}
33{,}523{,}800 \\
31{,}744{,}025 \\
11{,}576{,}590 \\
1{,}807{,}755 \\
15{,}275 \\
1 \ + \\
\hline
78{,}667{,}446
\end{array}
$$

StateとDP ValueとTrace-backの為に必要なバイト数 =
$78.7\times 10^6 \times 3 \times 4_{\text{バイト}} =$ 945 Mbytes.

States K と States (K+1) との間の checking 比較総回数 =
$1\times 33.6\times 10^6 + 33.6\times 31.8\times 10^{12} + 31.8\times 11.6\times 10^{12} + 11.6\times 1.81\times 10^{12}$
$+ 1.81\times 0.016\times 10^{12} + 15{,}275$
$= 33.6\times 10^6 + (1068.48 + 368.88 + 20.996 + 0.029)\times 10^{12} + 15{,}275$
$= 1458.385\times 10^{12} + 33.6\times 10^6 + 15{,}275 \approx 1459\times 10^{12}$ 回 $\approx 1.5\times 10^{15}$ 回
= 1459 Tera 回のcheckingを行なう。$= \sum_{K=0}^{5} NS(K) * NS(K+1)$ □

ここまで 何とかパソコンに 載せられる。□

# なぜR.E.JensenのNetworkの方がよいのかの理由 (36)

a. M=3, N=6 とか M=3, N=7 の時は distribution form 毎に DP 計算した方がメモリも少なくてよいし、計算も早い。

b. M=6, N=24 の時の distribution forms 表を見ると 199個のうち、連続して1が2個以上続くものは約半分(79こ)しかない。従って Singleton set だけを特別に扱って高速化するのはあまり効果がない。

c. M=6, N=24 に対し distribution form 毎に Network を作るとしてみる。

9 5 5 2 2 1 と 9 5 4 4 1 1 とを考えると

| | |
|---|---|
| $NS(1)=\binom{24}{9}$ ----① | $NS(1)=\binom{24}{9}$ ----③ |
| $NS(2)=\binom{24}{9+5}=\binom{24}{14}$ --② | $NS(2)=\binom{24}{9+5}=\binom{24}{14}$ --④ |
| $NS(3)=\binom{24}{19}=\binom{24}{5}$ , | $NS(3)=\binom{24}{18}$ |
| $NS(4)=\binom{24}{21}=\binom{24}{3}$ | $NS(4)=\binom{24}{22}$ |
| $NS(5)=\binom{24}{23}=24$ | $NS(5)=\binom{24}{23}$ |
| $NS(6)=1.$ | $NS(6)=1.$ |

9 5 5 2 2 1 に対する NS(1), NS(2) は 9 5 4 4 1 1 に対する NS(1), NS(2) と同一で各々の集合は1対1対応している。つまり同一の状態を distribution form 毎(ごと)に作ることになる。M, N がこれより各々大きくなれば非能率は更に高まるだろう。①=③, ②=④に注意。

d. 2003.12.15-IV の様な極限までの state の圧縮をしない限り Jensen の方法が優れている。M, N が増大した時でも極限までの state の圧縮は効果があるか?? 現在は不明。2003.12.21現在、Jensenの方法採用決定。

No. of distribution forms for n= 24 is 199, where M = 6 N = 24

k3/100 = 0

```
19 1 1 1 1 1     18 2 1 1 1 1     17 3 1 1 1 1     17 2 2 1 1 1     16 4 1 1 1 1
16 3 2 1 1 1     16 2 2 2 1 1     15 5 1 1 1 1     15 4 2 1 1 1     15 3 3 1 1 1
15 3 2 2 1 1     15 2 2 2 2 1     14 6 1 1 1 1     14 5 2 1 1 1     14 4 3 1 1 1
14 4 2 2 1 1     14 3 3 2 1 1     14 3 2 2 2 1     14 2 2 2 2 2     13 7 1 1 1 1
13 6 2 1 1 1     13 5 3 1 1 1     13 5 2 2 1 1     13 4 4 1 1 1     13 4 3 2 1 1
13 4 2 2 2 1     13 3 3 3 1 1     13 3 3 2 2 1     13 3 2 2 2 2     12 8 1 1 1 1
12 7 2 1 1 1     12 6 3 1 1 1     12 6 2 2 1 1     12 5 4 1 1 1     12 5 3 2 1 1
12 5 2 2 2 1     12 4 4 2 1 1     12 4 3 3 1 1     12 4 3 2 2 1     12 4 2 2 2 2
12 3 3 3 2 1     12 3 3 2 2 2     11 9 1 1 1 1     11 8 2 1 1 1     11 7 3 1 1 1
11 7 2 2 1 1     11 6 4 1 1 1     11 6 3 2 1 1     11 6 2 2 2 1     11 5 5 1 1 1
11 5 4 2 1 1     11 5 3 3 1 1     11 5 3 2 2 1     11 5 2 2 2 2     11 4 4 3 1 1
11 4 4 2 2 1     11 4 3 3 2 1     11 4 3 2 2 2     11 3 3 3 3 1     11 3 3 3 2 2
10 10 1 1 1 1    10 9 2 1 1 1     10 8 3 1 1 1     10 8 2 2 1 1     10 7 4 1 1 1
10 7 3 2 1 1     10 7 2 2 2 1     10 6 5 1 1 1     10 6 4 2 1 1     10 6 3 3 1 1
10 6 3 2 2 1     10 6 2 2 2 2     10 5 5 2 1 1     10 5 4 3 1 1     10 5 4 2 2 1
10 5 3 3 2 1     10 5 3 2 2 2     10 4 4 4 1 1     10 4 4 3 2 1     10 4 4 2 2 2
10 4 3 3 3 1     10 4 3 3 2 2     10 3 3 3 3 2     9 9 3 1 1 1      9 9 2 2 1 1
9 8 4 1 1 1      9 8 3 2 1 1      9 8 2 2 2 1      9 7 5 1 1 1      9 7 4 2 1 1
9 7 3 3 1 1      9 7 3 2 2 1      9 7 2 2 2 2      9 6 6 1 1 1      9 6 5 2 1 1
9 6 4 3 1 1      9 6 4 2 2 1      9 6 3 3 2 1      9 6 3 2 2 2      9 5 5 3 1 1
```

k3/100 = 1

```
9 5 5 2 2 1      9 5 4 4 1 1      9 5 4 3 2 1      9 5 4 2 2 2      9 5 3 3 3 1
9 5 3 3 2 2      9 4 4 4 2 1      9 4 4 3 3 1      9 4 4 3 2 2      9 4 3 3 3 2
9 3 3 3 3 3      8 8 5 1 1 1      8 8 4 2 1 1      8 8 3 3 1 1      8 8 3 2 2 1
8 8 2 2 2 2      8 7 6 1 1 1      8 7 5 2 1 1      8 7 4 3 1 1      8 7 4 2 2 1
8 7 3 3 2 1      8 7 3 2 2 2      8 6 6 2 1 1      8 6 5 3 1 1      8 6 5 2 2 1
8 6 4 4 1 1      8 6 4 3 2 1      8 6 4 2 2 2      8 6 3 3 3 1      8 6 3 3 2 2
8 5 5 4 1 1      8 5 5 3 2 1      8 5 5 2 2 2      8 5 4 4 2 1      8 5 4 3 3 1
8 5 4 3 2 2      8 5 3 3 3 2      8 4 4 4 3 1      8 4 4 4 2 2      8 4 4 3 3 2
8 4 3 3 3 3      7 7 7 1 1 1      7 7 6 2 1 1      7 7 5 3 1 1      7 7 5 2 2 1
7 7 4 4 1 1      7 7 4 3 2 1      7 7 4 2 2 2      7 7 3 3 3 1      7 7 3 3 2 2
7 6 6 3 1 1      7 6 6 2 2 1      7 6 5 4 1 1      7 6 5 3 2 1      7 6 5 2 2 2
7 6 4 4 2 1      7 6 4 3 3 1      7 6 4 3 2 2      7 6 3 3 3 2      7 5 5 5 1 1
7 5 5 4 2 1      7 5 5 3 3 1      7 5 5 3 2 2      7 5 4 4 3 1      7 5 4 4 2 2
7 5 4 3 3 2      7 5 3 3 3 3      7 4 4 4 4 1      7 4 4 4 3 2      7 4 4 3 3 3
6 6 6 4 1 1      6 6 6 3 2 1      6 6 6 2 2 2      6 6 5 5 1 1      6 6 5 4 2 1
6 6 5 3 3 1      6 6 5 3 2 2      6 6 4 4 3 1      6 6 4 4 2 2      6 6 4 3 3 2
6 6 3 3 3 3      6 5 5 5 2 1      6 5 5 4 3 1      6 5 5 4 2 2      6 5 5 3 3 2
6 5 4 4 4 1      6 5 4 4 3 2      6 5 4 3 3 3      6 4 4 4 4 2      6 4 4 4 3 3
5 5 5 5 3 1      5 5 5 5 2 2      5 5 5 4 4 1      5 5 5 4 3 2      5 5 5 3 3 3
5 5 4 4 4 2      5 5 4 4 3 3      5 4 4 4 4 3      4 4 4 4 4 4
```

1, 2, …, N (2≦N≦25) の中から k 個を選んで作られる全ての k 個よりなる {1, 2, …, N} の部分集合を作る int 型関数 Subsets( int␣N, int␣k, int␣tbl[ ], int␣ks ) (2≦k≦N)

a. 2≦k≦N と仮定する。

b. 各々の集合は int 型か unsigned␣int 型で保持する。集合の元を復元するには int 型のほうが早いと言えるか、本当に。

c. $2^{25}=2^{10}\cdot 2^{10}\cdot 2^{5}=$ 33,554,432 (≒ 3360万) ≈ 34×4 M bytes = 136 MBytes. (136M = $1.36\times 10^{8}$)
{1, 2, …, N} の全ての部分集合を 136 M bytes 使ってメモリに保持出来る。Singleton でない各部分集合に対し評価値を毎回計算することを止めて、テーブル探索することも出来る。この場合、各集合値からテーブルの番地を瞬時に対応させることは出来る。空集合、Singleton はこのテーブルに作らない。(N+1) 個を未使用にしておけばよい。未使用 (N+1) 個はテーブルの先頭から連続して (N+1) 個続くのではないことに注意せよ。State と DP Value 保持双方の為に必要なバイト数 = 136 M bytes + 136 M bytes (単精度実数) = 272 M bytes.

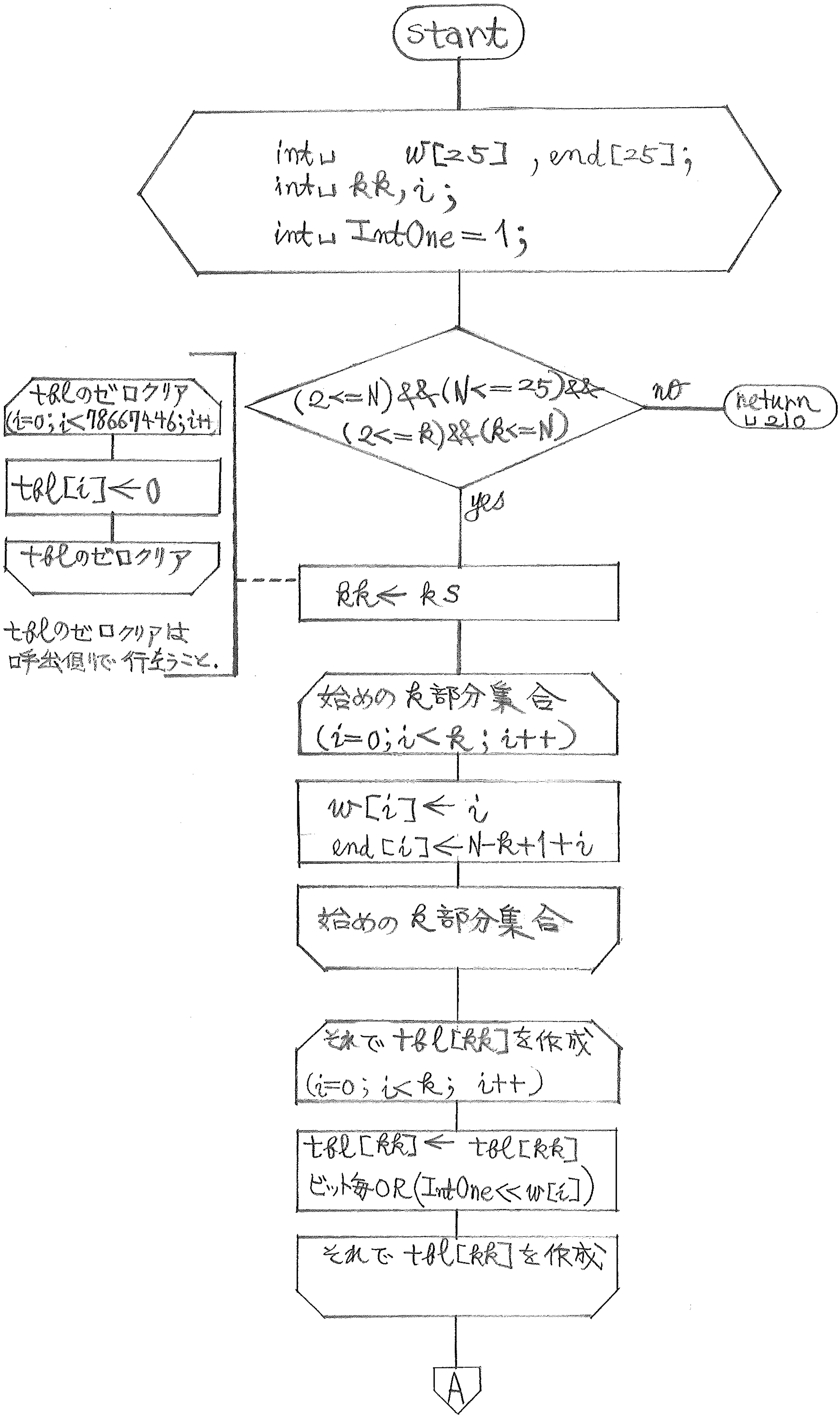
d. 入力データ：N, k ; 部分集合を作るテーブル名 tbl，部分集合を作り始めるテーブル内番地 ks.

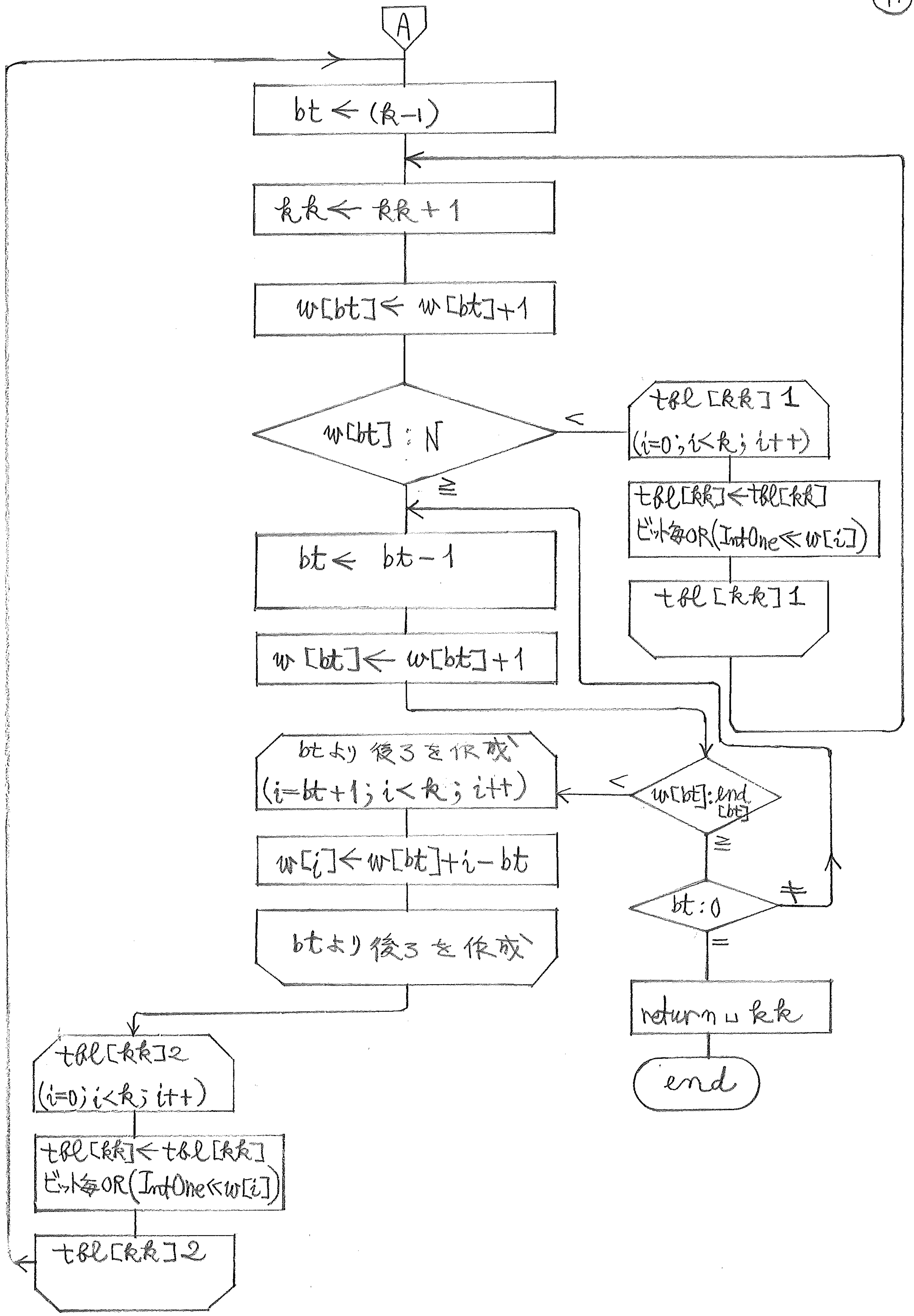
e. 出力データ：CNk $\left(={}_{n}C_{k}=\binom{n}{k}\right)$，部分集合を作り終った時のテーブル内番地 ke，部分集合 tbl[kk] (kk = ks から ke).

f. テーブル tbl のゼロクリアは呼出側で行うこと。

# データ部の定義　intu Subsets

| 変数名または配列名 | データの型、配列の大きさ | 意味 |
|---|---|---|
| N | int, local | 入力データ。元となる集合{1,2,…,N}の大きさ, 2≦N≦25とする。 |
| k | int, local | 入力データ。作る部分集合の要素数, 作る部分集合の大きさ, $_NC_k$. |
| tbl | int, [78 667 446] | k個よりなる{1,2,…,N}の全ての部分集合を入れる配列, 大きさはM,Nに依存する。N=25, M=6のとき 78667446である。 |
| ks | int | 部分集合を作り始める配列tblのテーブル内番地. |
| ke | int | 部分集合を作り終った時の配列tblのテーブル内番地. 結果的に不使用となる。 |
| w ✓ | int, local [25] | 部分集合を作るための作業用配列。大きさ 25. |
| kk ✓ | int | 制御変数 |
| i ✓ | int | 制御変数. |
| IntOne ✓ | int | 定数, 固定小数点数の1(ハち). |
| bt ✓ | int | back track 変数 |
| end ✓ | int, [25] | back track 制御用配列 |


start
int␣ w[25] , end[25];
int␣ kk, i;
int␣ IntOne = 1;
(2<=N)&&(N<=25)&&
(2<=k)&&(k<=N)
no
return
␣210
yes
tblのゼロクリア
(i=0; i<78667446; i++)
tbl[i] ← 0
tblのゼロクリア
tblのゼロクリアは
呼出側で行なうこと.
kk ← ks
始めのk部分集合
(i=0; i<k; i++)
w[i] ← i
end[i] ← N-k+1+i
始めのk部分集合
それでtbl[kk]を作成
(i=0; i<k; i++)
tbl[kk] ← tbl[kk]
ビット毎OR(IntOne<<w[i])
それでtbl[kk]を作成
A


A
bt ← (k−1)
kk ← kk + 1
w[bt] ← w[bt]+1
w[bt] : N
<
≧
tbl[kk]1
(i=0; i<k; i++)
tbl[kk] ← tbl[kk]
ビット毎OR(IntOne ≪ w[i])
tbl[kk]1
bt ← bt − 1
w[bt] ← w[bt]+1
btより後ろを作成
(i=bt+1; i<k; i++)
<
w[bt]:end[bt]
≧
w[i] ← w[bt]+i−bt
btより後ろを作成
bt : 0
≠
=
return␣kk
end
tbl[kk]2
(i=0; i<k; i++)
tbl[kk] ← tbl[kk]
ビット毎OR(IntOne ≪ w[i])
tbl[kk]2

# $N=6$; $k=2,3,4,5$の時の$\{0,1,2,3,4,\overset{N-1}{\overset{\|}{5}}\}$の部分集合, デバック用

## データ

| 0 | 1 |
|---|---|
| 0 | 1 |
| 0 | 2 |
| 0 | 3 |
| 0 | 4 |
| 0 | 5 |
| 1 | 2 |
| 1 | 3 |
| 1 | 4 |
| 1 | 5 |
| 2 | 3 |
| 2 | 4 |
| 2 | 5 |
| 3 | 4 |
| 3 | 5 |
| 4 | 5 |
| 5 | 6 |

$k=2$, $\binom{6}{2}=15$ 通り

| 0 | 1 | 2 |
|---|---|---|
| 0 | 1 | 2 |
| 0 | 1 | 3 |
| 0 | 1 | 4 |
| 0 | 1 | 5 |
| 0 | 2 | 3 |
| 0 | 2 | 4 |
| 0 | 2 | 5 |
| 0 | 3 | 4 |
| 0 | 3 | 5 |
| 0 | 4 | 5 |
| 1 | 2 | 3 |
| 1 | 2 | 4 |
| 1 | 2 | 5 |
| 1 | 3 | 4 |
| 1 | 3 | 5 |
| 1 | 4 | 5 |
| 2 | 3 | 4 |
| 2 | 3 | 5 |
| 2 | 4 | 5 |
| 3 | 4 | 5 |
| 4 | 5 | 6 |

$k=3$, $\binom{6}{3}=20$ 通り

| 0 | 1 | 2 | 3 |
|---|---|---|---|
| 0 | 1 | 2 | 3 |
| 0 | 1 | 2 | 4 |
| 0 | 1 | 2 | 5 |
| 0 | 1 | 3 | 4 |
| 0 | 1 | 3 | 5 |
| 0 | 1 | 4 | 5 |
| 0 | 2 | 3 | 4 |
| 0 | 2 | 3 | 5 |
| 0 | 2 | 4 | 5 |
| 0 | 3 | 4 | 5 |
| 1 | 2 | 3 | 4 |
| 1 | 2 | 3 | 5 |
| 1 | 2 | 4 | 5 |
| 1 | 3 | 4 | 5 |
| 2 | 3 | 4 | 5 |
| 3 | 4 | 5 | 6 |

$k=4$, $\binom{6}{4}=15$ 通り

| 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 |
| 0 | 1 | 2 | 3 | 5 |
| 0 | 1 | 2 | 4 | 5 |
| 0 | 1 | 3 | 4 | 5 |
| 0 | 2 | 3 | 4 | 5 |
| 1 | 2 | 3 | 4 | 5 |
| 2 | 3 | 4 | 5 | 6 |

$k=5$, $\binom{6}{5}=6$ 通り

15
20
15
6

# Subsets(int␣N, int␣k, int␣tbl[], int␣ks, int␣ke)のデバック

| N k ks | kk | bt | i=bt+1~(k-1) | | 配列 w 0 | 1 | 2 | 3 | 4 | 配列 tblのビット 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 6 2 0 | 0 | | | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | |
| | | | | | 0 | 1 | | | | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| | 1 | 1 | | | | 2 | | | | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| | 2 | | | | | 3 | | | | 0 | 0 | 1 | 0 | 0 | 1 | 2 |
| | 3 | | | | | 4 | | | | 0 | 1 | 0 | 0 | 0 | 1 | 3 |
| | 4 | | | | | 5 | | | | 1 | 0 | 0 | 0 | 0 | 1 | 4 |
| | 5 | | | | | 6 | | | | | | | | | | |
| | | 0 | 1~1 | | 1 | 2 | | | | 0 | 0 | 0 | 1 | 1 | 0 | 5 |
| | 6 | 1 | | | | 3 | | | | 0 | 0 | 1 | 0 | 1 | 0 | 6 |
| | 7 | | | | | 4 | | | | 0 | 1 | 0 | 0 | 1 | 0 | 7 |
| | 8 | | | | | 5 | | | | 1 | 0 | 0 | 0 | 1 | 0 | 8 |
| | 9 | | | | | 6 | | | | | | | | | | |
| | | 0 | 1~1 | | 2 | 3 | | | | 0 | 0 | 1 | 1 | 0 | 0 | 9 |
| | 10 | 1 | | | | 4 | | | | 0 | 1 | 0 | 1 | 0 | 0 | 10 |
| | 11 | | | | | 5 | | | | 1 | 0 | 0 | 1 | 0 | 0 | 11 |
| | 12 | | | | | 6 | | | | | | | | | | |
| | | 0 | 1~1 | | 3 | 4 | | | | 0 | 1 | 1 | 0 | 0 | 0 | 12 |
| | 13 | 1 | | | | 5 | | | | 1 | 0 | 1 | 0 | 0 | 0 | 13 |
| | 14 | | | | | 6 | | | | | | | | | | |
| | | 0 | 1~1 | | 4 | 5 | | | | 1 | 1 | 0 | 0 | 0 | 0 | 14 |
| | 15 | 1 | | | | 6 | | | | | | | | | | |
| | | 0 | | | 5 | | | | | | | | | | | |
| | | | | 正しい。 | | | 2003. | 12. | 25. | Check | 完 | 了。 | | | | |

| N k ks | kk | bt | i=bt+1~(k-1) | | w 0 | 1 | 2 | 3 | 4 | tbl 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 6 5 15 | 15 | | | | 0 | 1 | 2 | 3 | 4 | 0 | 1 | 1 | 1 | 1 | 1 | 15 |
| | 16 | 4 | | | | | | | 5 | 1 | 0 | 1 | 1 | 1 | 1 | 16 |
| | 17 | | | | | | | | 6 | | | | | | | |
| | | 3 | 4~4 | | | | | 4 | 5 | 1 | 1 | 0 | 1 | 1 | 1 | 17 |
| | 18 | 4 | | | | | | | 6 | | | | | | | |
| | | 3 | | | | | | 5 | | | | | | | | |
| | | 2 | 3~4 | | | | 3 | 4 | 5 | 1 | 1 | 1 | 0 | 1 | 1 | 18 |
| | | | | | | | | | 6 | | | | | | | |
| | 19 | 4 | | | | | | | | | | | | | | |
| | | 3 | | | | | | 5 | | | | | | | | |
| | | 2 | | | | | 4 | | | | | | | | | |
| | | 1 | 2~4 | | | 2 | 3 | 4 | 5 | 1 | 1 | 1 | 1 | 0 | 1 | 19 |
| | | | | | | | | | 6 | | | | | | | |
| | 20 | 4 | | | | | | | | | | | | | | |
| | | 3 | | | | | | 5 | | | | | | | | |
| | | 2 | | | | | 4 | | | | | | | | | |
| | | 1 | 2~4 | | | 3 | | | | | | | | | | |
| | | 0 | 1~4 | | 1 | 2 | 3 | 4 | 5 | 1 | 1 | 1 | 1 | 1 | 0 | 20 |
| | | | | | | | | | 6 | | | | | | | |
| | 21 | 4 | | | | | | 5 | | | | | | | | |
| | | 3 | | | | | | | | | | | | | | |
| | | 2 | | | | | 4 | | | | | | | | | |
| | | 1 | | | | 3 | | | | | | | | | | |
| | | 0 | | 正しい。 | 2 | | | | | | | | | | | |

M, Nの値によってNS(K)（$1 \leqq K \leqq M$）の値が違うので、$2 \leqq N \leqq 25$ かつ

$2 \leqq M \leqq N$ の条件下でのNS(K)の値をパスカルの三角形（2

項係数表）とJensenのmin(K), max(K)（$1 \leqq K \leqq M$）より作成する。

int␣Prepare_Network(int␣M, int␣N, int␣NS[ ],

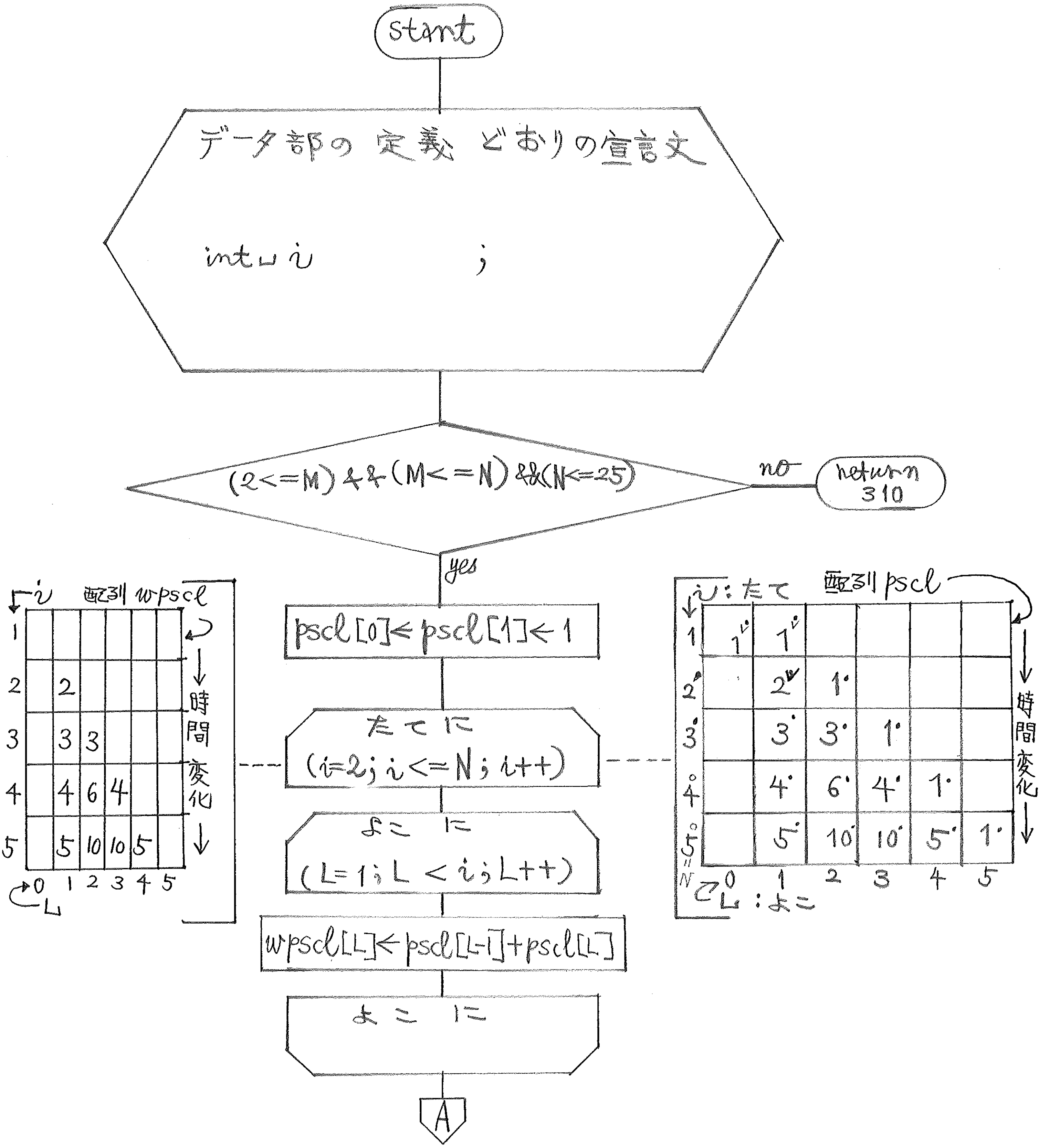
int␣min[ ], int␣max[ ]　　　　)

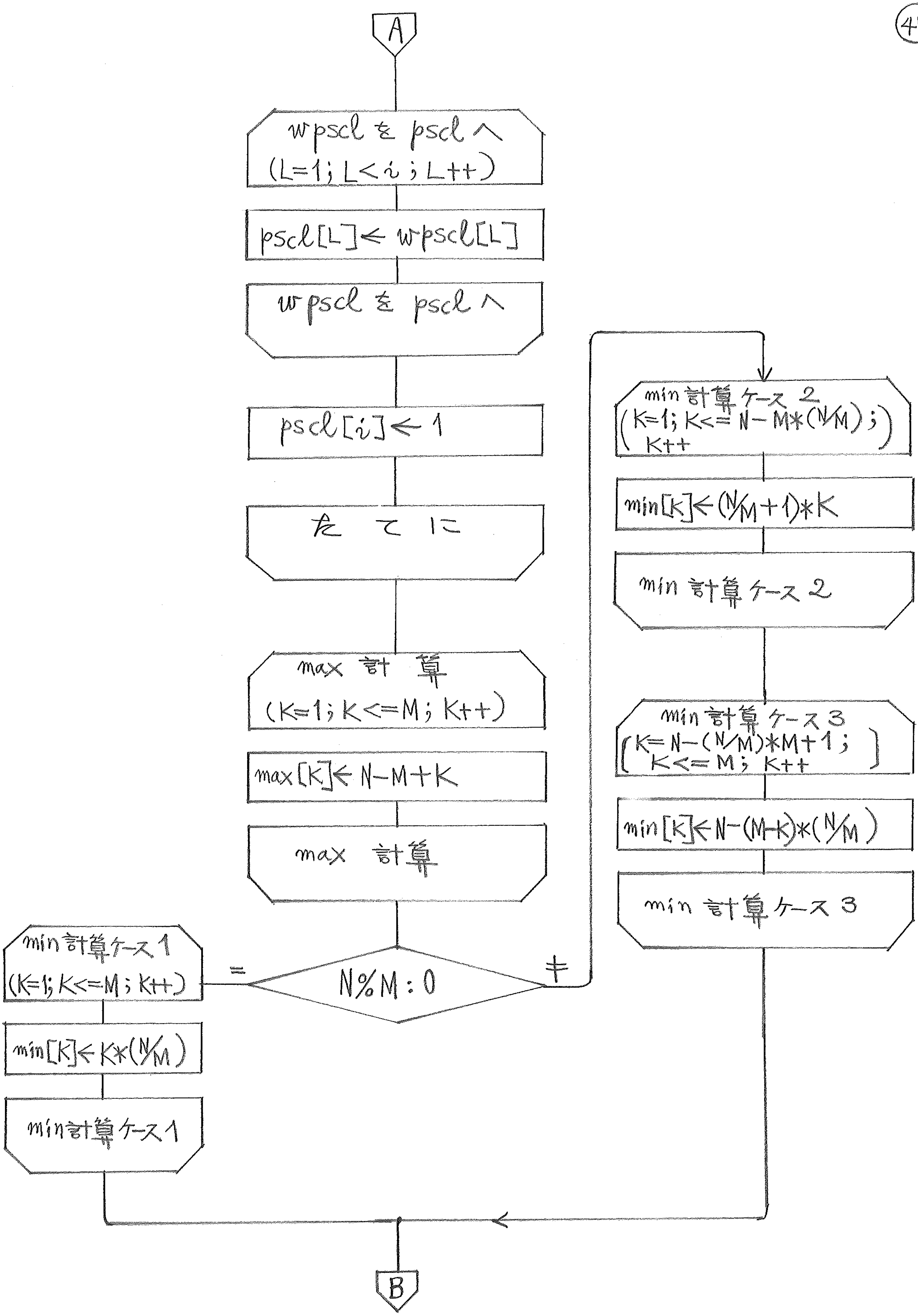
# データの定義

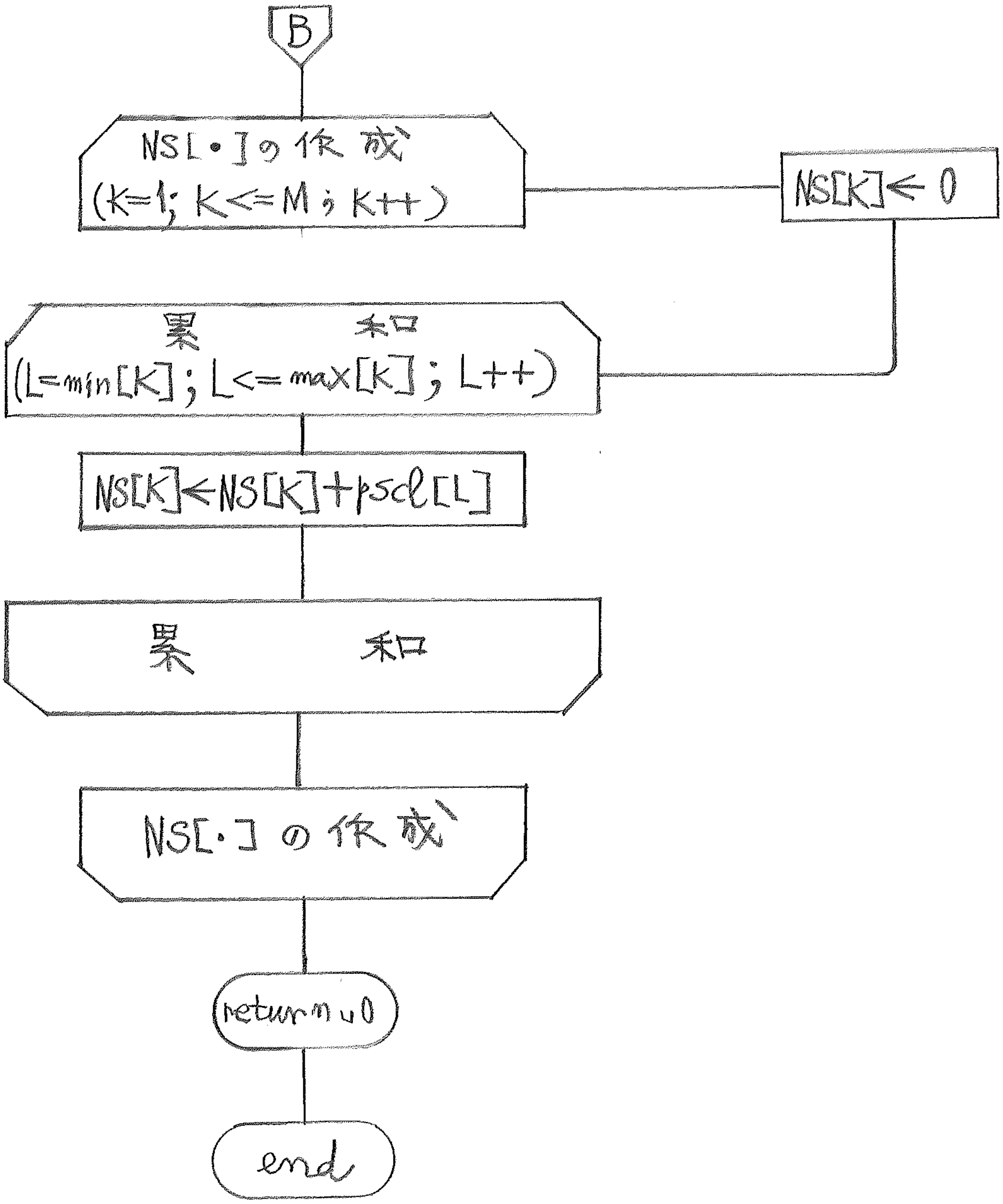
| 変数名または配列名 | データの型、配列の大きさ | 意味 |
|---|---|---|
| M | int | 分割数，$2 \leqq M \leqq N \leqq 25$。入力データ. |
| N | int | データ総数，$2 \leqq N \leqq 25$. 入力データ. |
| M0 | int | Jensen の paper の $M_0$，使うかどうかはよく考えて。常に $M0 \equiv M$ とおいてもよいのだから。常に $M0 = M$ とする。 |
| NS | unsigned int, [25+1] | Stage K における state 総数を NS[K] に入れる。$NS[K] = \sum_{L=min(k)}^{max(K)} \binom{N}{L}$。出力データ |
| pscl | unsigned int, [25+1] | $pscl[L] = \binom{N}{L} \quad (0 \leqq L \leqq N)$ pscl comes from Pascal table. |
| wpscl | unsigned int, [25+1] | pscl[L] 計算用ワークエリア |
| max | short int, [25+1] | $max[K] = N-M+K \quad (1 \leqq K \leqq M)$ 出力データ，Jensen の paper を見よ。 |
| min | short int, [25+1] | $min[K] = \begin{cases} (N/M+1)*K \text{ for } 1 \leqq K \leqq N-M*(N/M), \\ N-(M-K)*(N/M), \text{ for } N-M*(N/M) < K \leqq M, \\ \quad M\text{が}N\text{を割り切らないとき.} \\ K*(N/M),\ M\text{が}N\text{を割切るとき.} \end{cases}$ 出力データ，Jensen の paper を見よ。 |
| L | int | 制御変数 |
| K | int | 制御変数 |

関数 int␣Prepare-Network(int␣M,␣int␣N, int␣NS[], int␣M0,

int␣min[], int␣max[])




A
wpscl を pscl へ
(L=1; L<i; L++)
pscl[L] ← wpscl[L]
wpscl を pscl へ
pscl[i] ← 1
たてに
max 計算
(K=1; K<=M; K++)
max[K] ← N−M+K
max 計算
N%M : 0
=
≠
min 計算ケース1
(K=1; K<=M; K++)
min[K] ← K*(N/M)
min 計算ケース1
min 計算ケース2
(K=1; K<=N−M*(N/M); K++)
min[K] ← (N/M+1)*K
min 計算ケース2
min 計算ケース3
(K=N−(N/M)*M+1; K<=M; K++)
min[K] ← N−(M−K)*(N/M)
min 計算ケース3
B


B
NS[・]の作成
(K=1; K<=M; K++)
NS[K]← 0
累 和
(L=min[K]; L<=max[K]; L++)
NS[K]←NS[K]+pscl[L]
累 和
NS[・]の作成
return␣0
end

Fuzzy Metric Clustering and Dynamic Programming

のための Cプログラム FMCDP.cpp

# データの定義

| 変数名または配列名 | データの型、配列の大きさ | 意味 |
|---|---|---|
| M | int | 分割数，2≦M≦N≦25，入力データ |
| N | int | データ総数，2≦N≦25，入力データ |
| NS | unsigned int, [ ] | Stage Kにおける Stage総数≡NS[K]，入力データ |
| state[K] | int 型 pointer, NS[K] | Stage K に属する全ての states (一次元表現される) の先頭アドレスを持つ。0≦K≦M。int **state ; ダイナミックアロケーションを使う。 |
| dpv[K] | float 型 pointer, NS[K] | Stage K に属する各 state の持つ DP value の先頭アドレスを持つ。0≦K≦M。float **dpv ; ダイナミックアロケーションを使う。 |
| al | int 型 | $\alpha = \frac{al}{100}$ $(1 \leqq al \leqq 100)$ |
| K | int | 制御変数 |
| k | int | 制御変数 |
| trace_back[K] | int 型 pointer, NS[K] | Stage K のある state の最適 dpv が Stage (K-1) のどの state から来たかの state 情報を持つ |
| d[ ][ ] | float 型 2次元配列 | float d[NMAX][NMAX] ; $\alpha = al/100.0$ に無関係に $d_{ij}(d/100.0)$ 用に d[i][j] を使う (1≦al≦100) |
| 一次元配列 min[ ] | short int, [25+1] | Pre-Network で作成，Jensen の paper を見よ。 |
| 一次元配列 max[ ] | short int, [25+1] | Pre-Network で作成，Jensen の paper を見よ。 |

N=25, M=6 の時 NS(1)=33,523,800. 2次元配列データの転送法についてはソースプログラムリストを見ること。

| | | |
|---|---|---|
| state_tbl | int 型, [33 554 432] | table 形式で持った state，2元以上の要素からなる state，実際は不要となろう。 |
| eval | float 型 [33554 432] | table 形式で持った eval，の要素からなる state 全てに対して計算する。 |

$33554432 = 2^{25} \rightarrow 33.6 \times 4$ M bytes ≒ 135 M bytes

# データの定義　　プログラム FMCDP

| 変数名または配列名 | データの型、配列の大きさ | 意味 |
|---|---|---|
| i | int | 制御変数 |
| j | int | 制御変数 |
| adn | float, [MMAX][ldmax] | adn[i][k] $= \underline{a}_{ik}$ $(0 \leqq k < ld)$, $0 \leqq i < N$ |
| aup | float, [MMAX][ldmax] | aup[i][k] $= \overline{a}_{ik}$ $(0 \leqq k < ld)$, $0 \leqq i < N$ |
| ld | int | l␣dimension, $\underline{a}_{ik}$ の k の種類 |
| dummy | int | adn[i][k], aup[i][k] を読む時の不要整数値 |
| kk | int | 制御変数 |
| ds | int | 差集合(diference set) = state[K][kk] \ state[K-1][k], ただし state[K-1][k] ⫋ state[K][kk] とする. |
| is | int | intersection of state[K-1][k] and state[K][kk] |
| stateKm1k | int | state[K-1][k] |
| ks | int | 制御変数 |
| pbname | char, [101] | 問題名(problem␣name, 100文字以内) |
| | | |

# デバック用テストデータ：H. Kamimura & M. Kurano, Clustering by a fuzzy metric, Fuzzy Sets and Systems 120 (2001) 249–254

| $i \backslash j$ | (0) 1 | (1) 2 | (2) 3 | (3) 4 | (4) 5 |
|---|---|---|---|---|---|
| 1 (0) | | [13, 14] | [32, 33] | [18, 19] | [1, 2] |
| 2 (1) | | | [5, 8] | [1, 4] | [8, 11] |
| 3 (2) | | | | [2, 3] | [25, 26] |
| 4 (3) | | | | | [13, 14] |
| 5 (4) | | | | | |

$\tilde{d}_{ij} = \tilde{d}_{ji}$, ; $\{1, 2, 3, 4, 5\}$, $M = 3$, $N = 5$, $\ell = 1$:

使用不可を発見 2003. 1. 14.

InCluster.txt の作成

```
     問題名(100文字以内)↵
     3␣␣␣␣␣␣5␣␣␣␣␣1↵
       1 行 空 白
i=0  0␣␣␣␣␣␣13␣14↵
            ␣32␣33↵
            ␣18␣19↵
            ␣1␣2↵
i=1  1␣␣␣␣␣5␣8↵
            ␣1␣4↵
            ␣8␣11↵
i=2  2␣␣␣␣␣2␣3↵
            ␣25␣26↵
i=3  3␣␣␣␣␣13␣14↵
```

OutCluster.txt への出力

```
 問題名(100文字以内)
M␣=␣3␣N␣=␣5␣ld␣=␣1↵
    1 行 空 白
    1 行 空 白
alpha          partition
    1 行 空 白
0.010000␣␣␣{1,␣5},␣{2,␣4},␣{3}↵
0.020000␣␣␣{1,␣5},␣{2,␣4},␣{3}↵
  ⁝
0.490000␣␣␣{1,␣5},␣{2,␣4},␣{3}↵
0.500000␣␣␣{1,␣5},␣{2,␣4},␣{3} または
           {1,␣5},␣{3,␣4},␣{2} のどちらか↵
0.510000␣␣␣{1,␣5},␣{3,␣4},␣{2}↵
  ⁝
1.000000␣␣␣{1,␣5},␣{3,␣4},␣{2}↵
```

# デバック用テストデータ：新規作成 2004.1.14 by K.Iwamura

| i \ l | 1 (0) | 2 (1) | | | |
|---|---|---|---|---|---|
| 1 (0) | [13, 14] | [32, 33] | | | |
| 2 (1) | [18, 19] | [1, 2] | | | |
| 3 (2) | [5, 8] | [1, 4] | | | |
| 4 (3) | [8, 11] | [2, 3] | | | |
| 5 (4) | [25, 26] | [13, 14] | | | |

$M=3$, $N=5$, $l=2$ で $\{1, 2, 3, 4, 5\}$ を3つに分割する。

## InCluster.txtの作成

```
問題データ名(100文字以内)
3␣5␣2⏎
 1行空白行
0␣␣␣␣␣␣13␣14␣␣32␣33⏎
1␣␣␣␣␣␣18␣19␣␣1␣2⏎
2␣␣␣␣␣␣5␣8␣␣1␣4⏎
3␣␣␣␣␣␣8␣11␣␣2␣3⏎
4␣␣␣␣␣␣25␣26␣␣13␣14⏎
```

## OutCluster.txtへの出力

```
 問題データ名(100文字以内)⏎
M␣=␣3␣N␣=␣5␣ld␣=␣2⏎
      2行空白
alpha␣=␣0.010000␣min.␣DP␣value␣=␣XXXXXX␣partition␣=
      1行空白
alpha␣=␣0.020000␣min.␣DP␣value␣=␣XXXXXX␣partition␣=␣
      1行空白
alpha␣=␣0.030000␣min.␣DP␣value␣=␣XXXXXX␣partition␣=␣
      1行空白
   ⁞
      1行空白
alpha␣=␣0.990000␣min.␣DP␣value␣=␣XXXXXX␣partition␣=␣
 空白1行
alpha␣=[illegible].000000␣min.␣DP␣value␣=␣XXXXXX␣partition␣=␣
```

# デバック用テストデータ：2004.1.15作成

$N=4, M=2, l=l_d=2.$ $\{0,1,2,3\}$を2つに分割する。 $N=4 \geqq 2 \cdot 2=2M.$ $M_0=M=2.$

| $i$ \ $l$ | 1 (0) | 2 (1) |
|---|---|---|
| 1 (0) | [0, 1] | [1, 2] |
| 2 (1) | [0, 2] | [1, 3] |
| 3 (2) | [0, 3] | [1, 4] |
| 4 (3) | [0, 4] | [2, 3] |

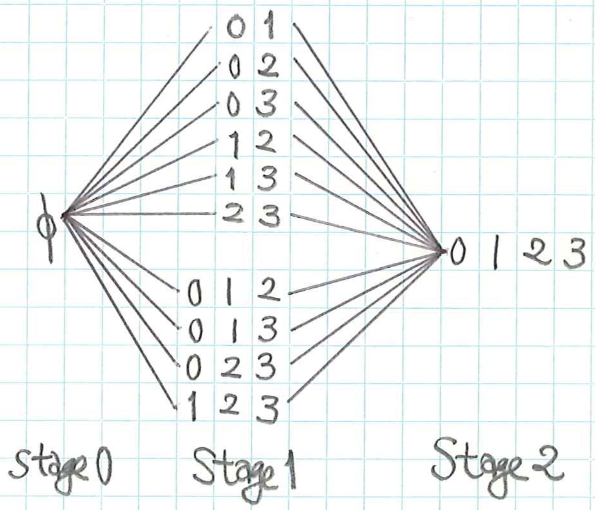
$max(1)=4-2+1=3$

$max(2)=4-2+2=4$

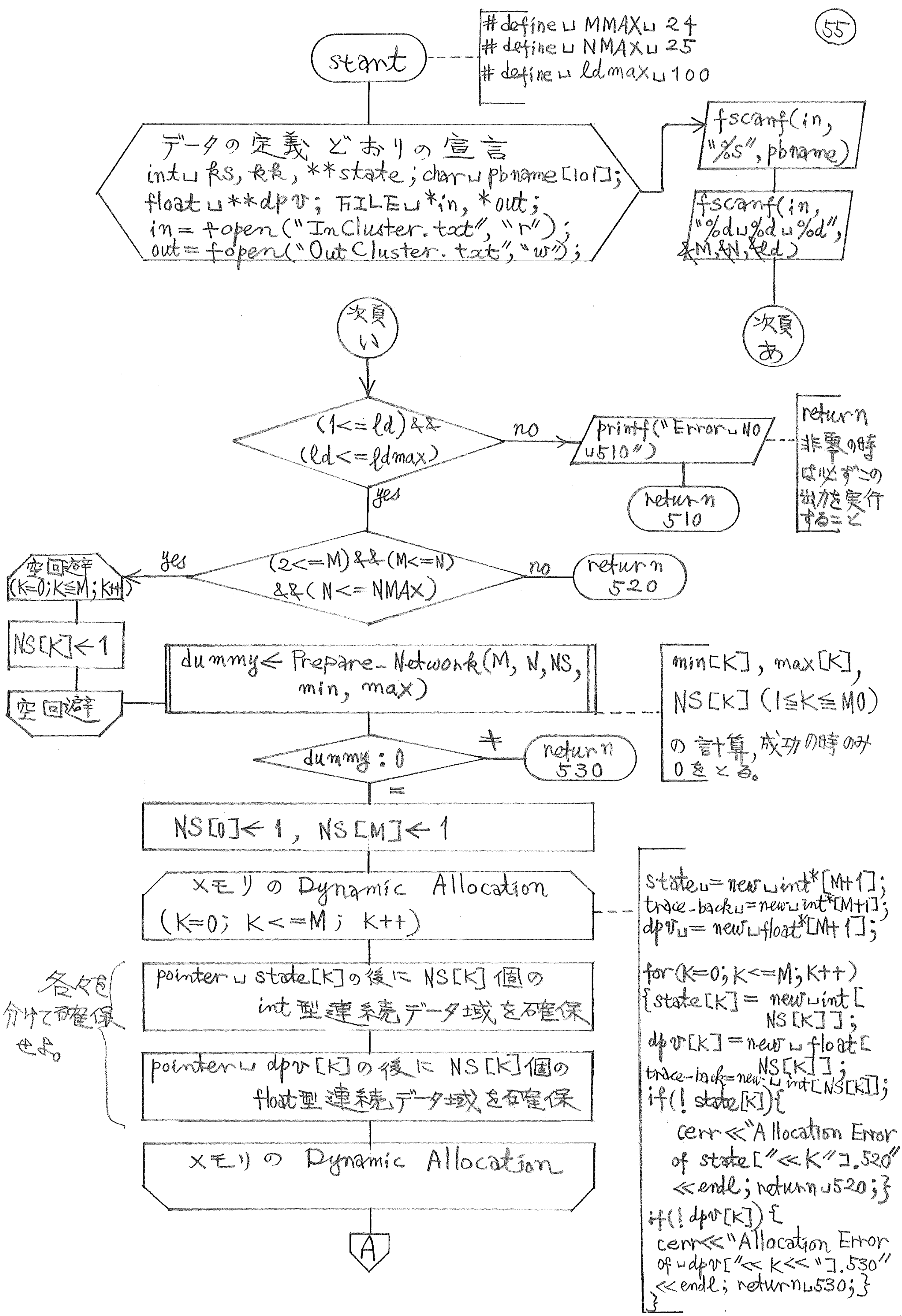
$min(1)=2$

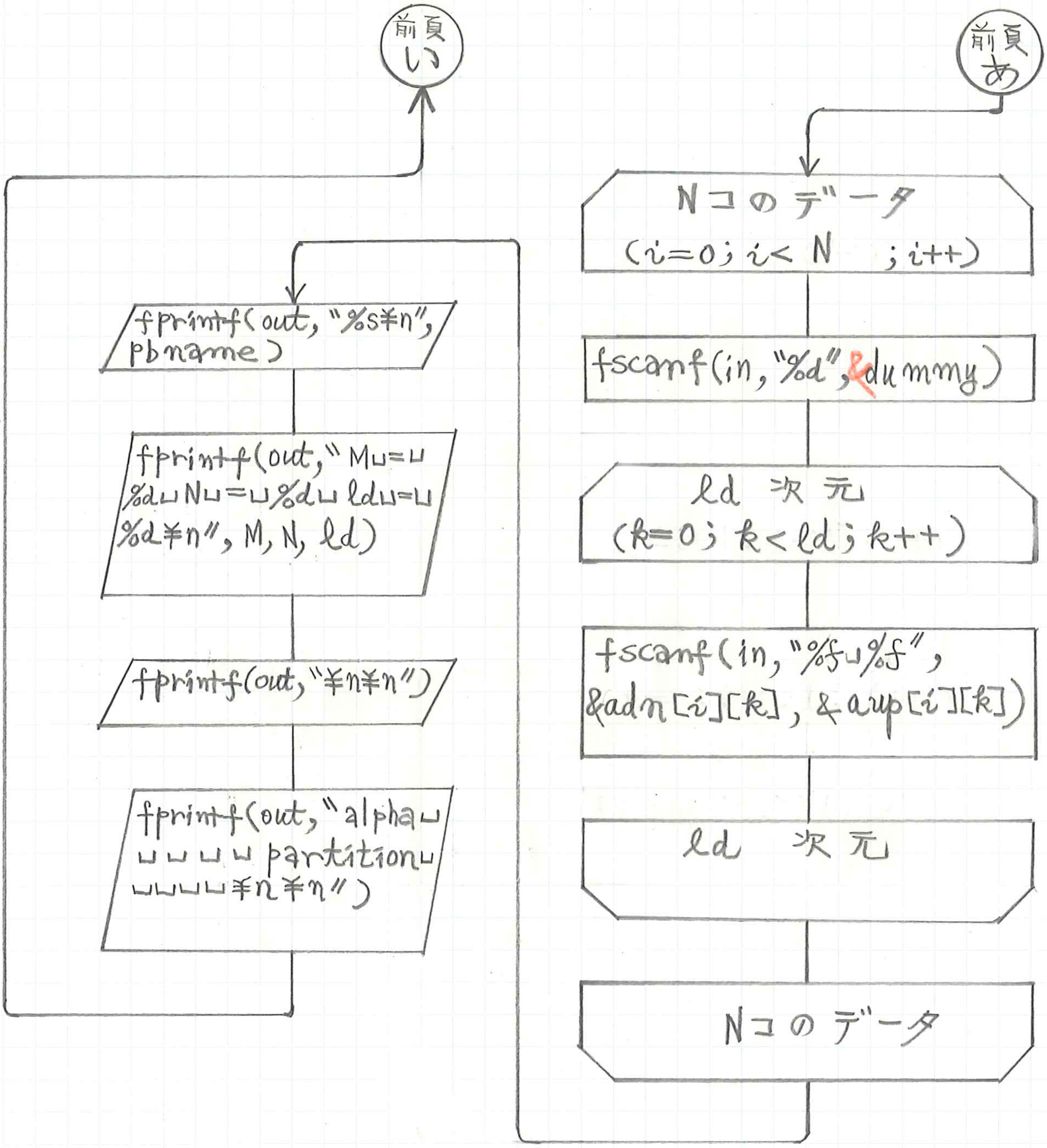
$min(2)=4$

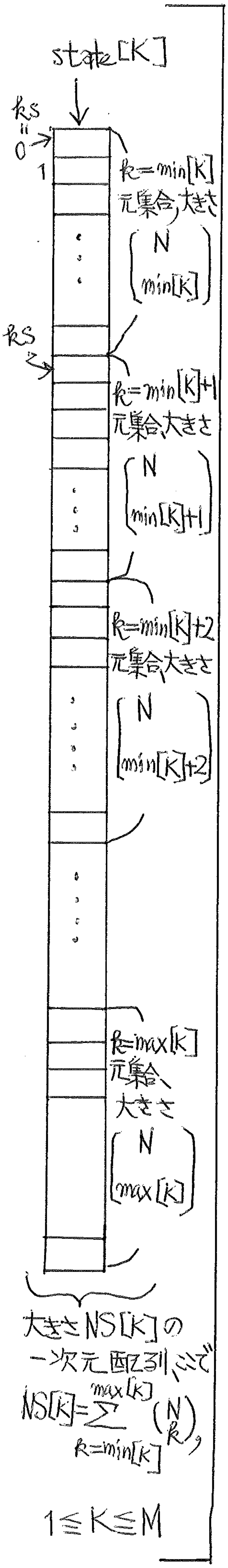
$NS(0)=1,\ NS(1)=\sum_{L=2}^{3}\binom{4}{L}=\binom{4}{2}+\binom{4}{3}$
$=\frac{4\cdot 3}{2}+\binom{4}{1}=6+\frac{4}{1}=10,\ NS(2)=$
$\binom{4}{4}=1.$

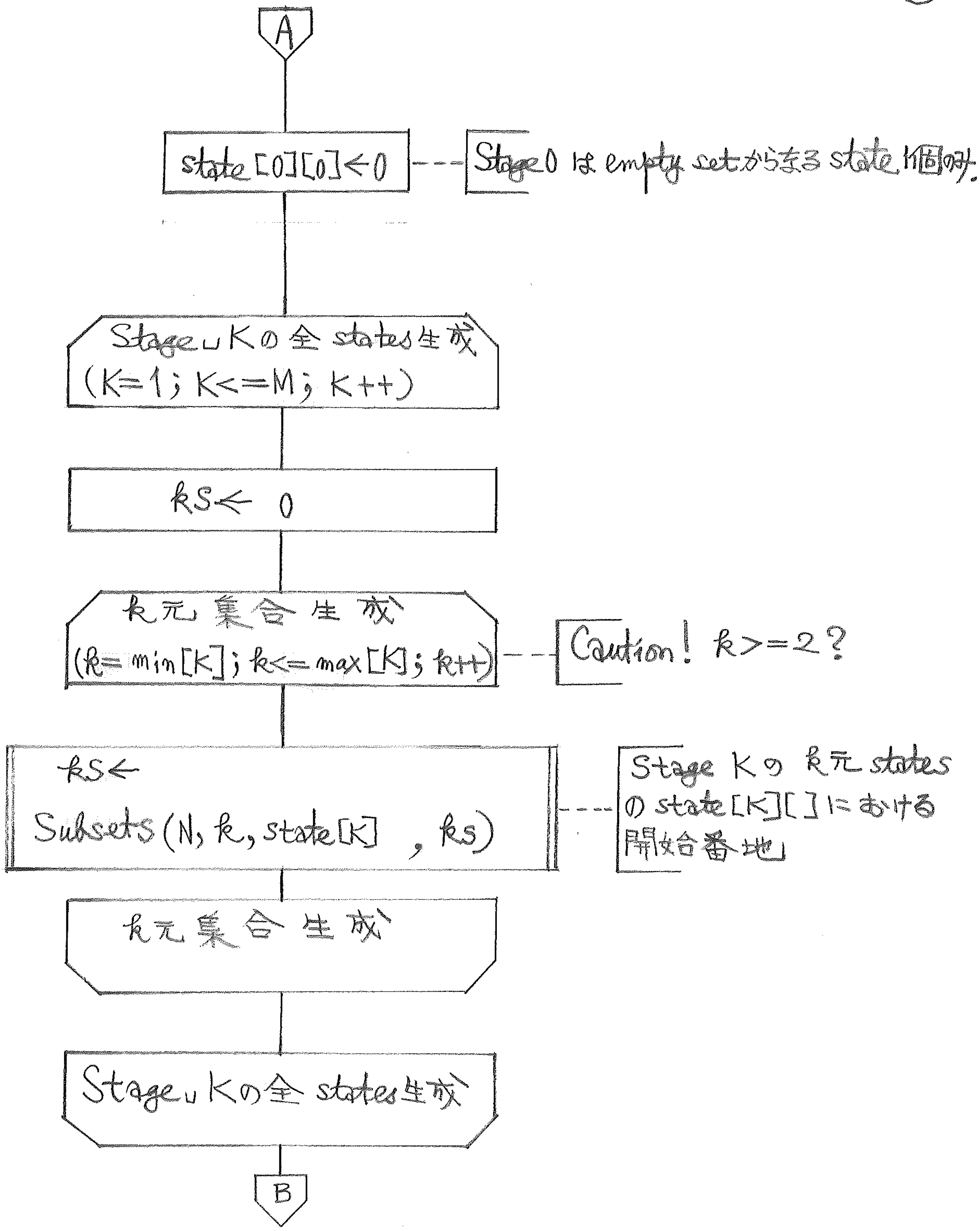


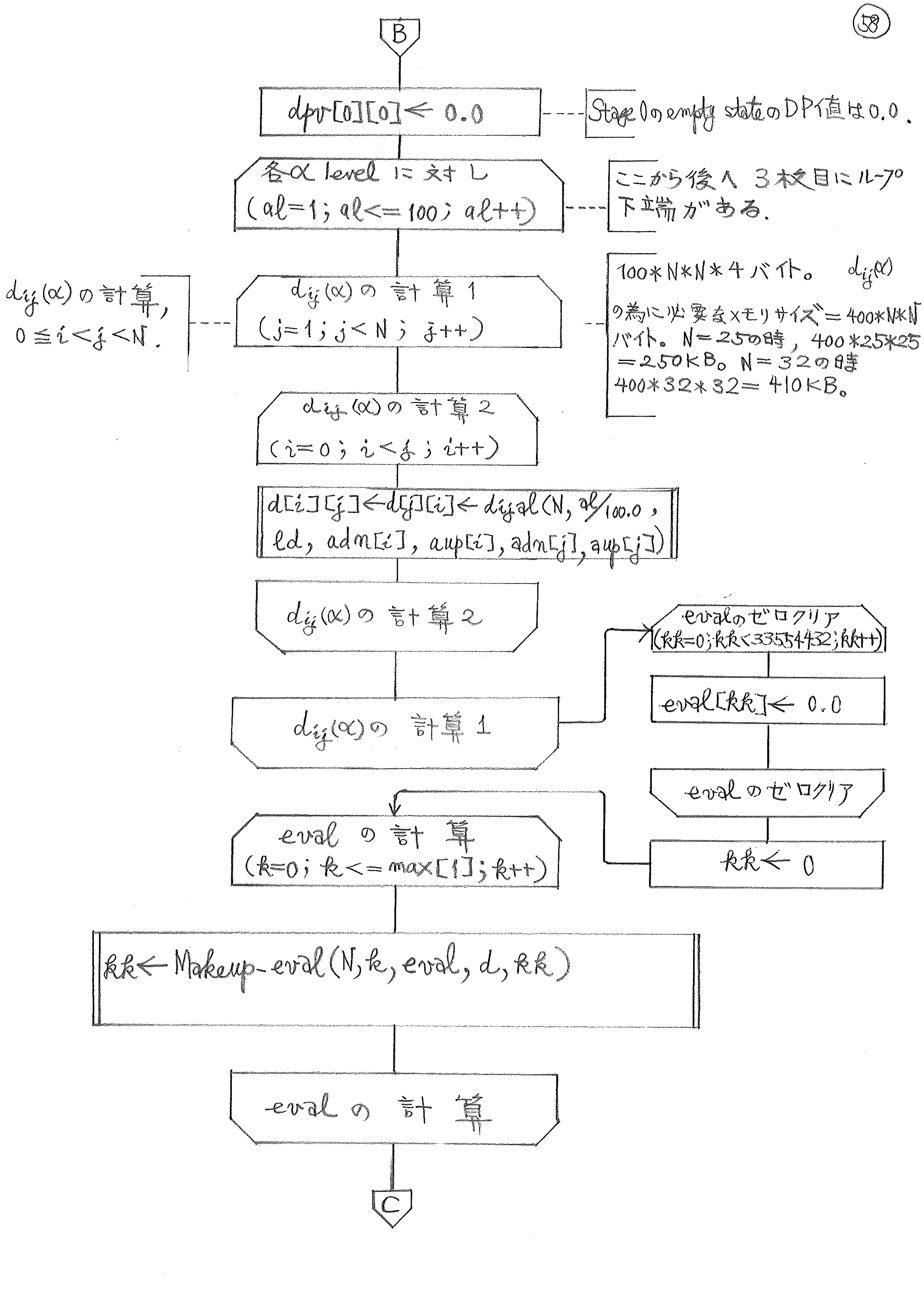

3元 state にのみ Fuzzy Metric のチェックが出来る。

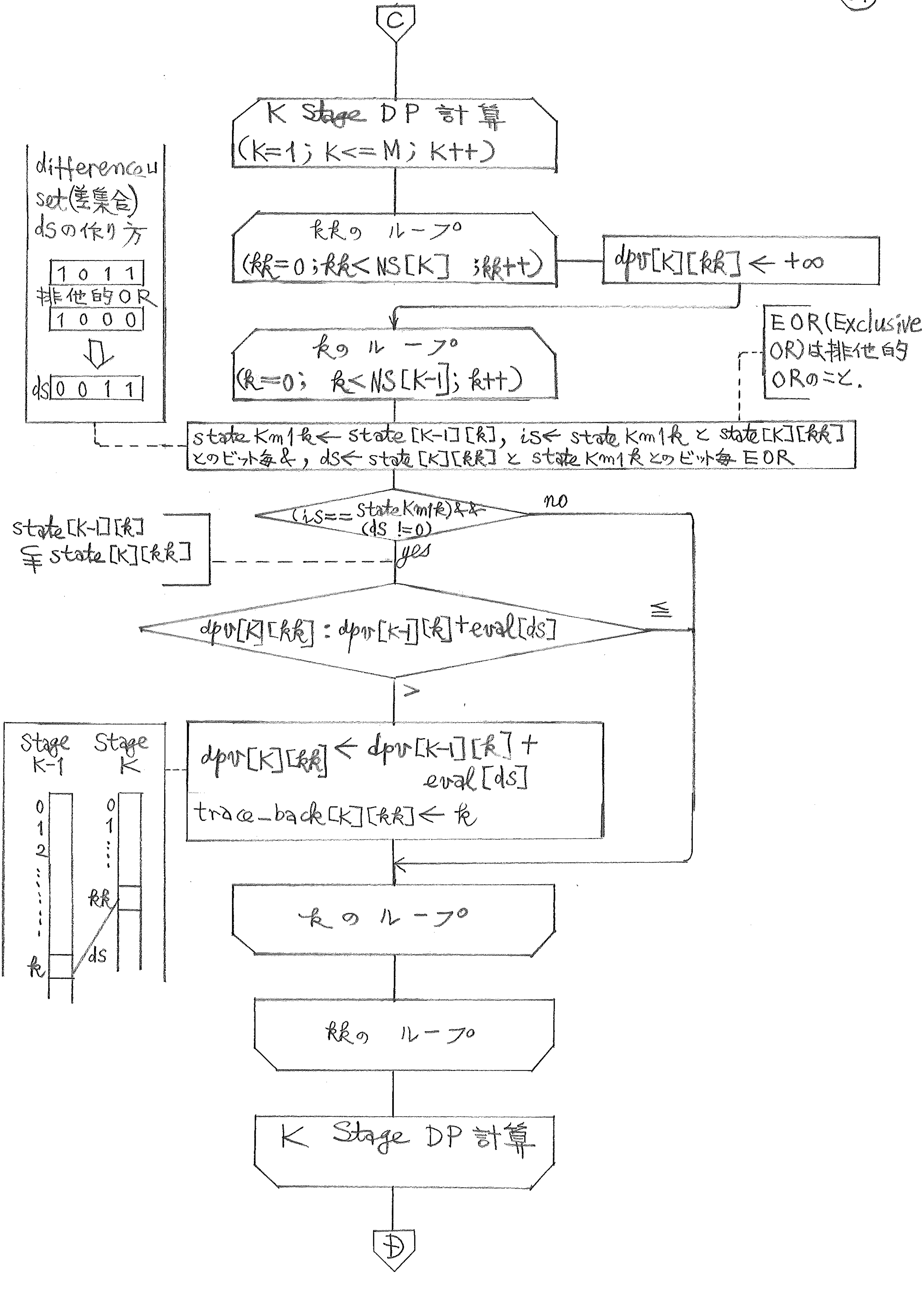

start
#define␣MMAX␣24
#define␣NMAX␣25
#define␣ldmax␣100
データの定義どおりの宣言
int␣ks, kk, **state; char␣pbname[101];
float␣**dpv; FILE␣*in, *out;
in=fopen("InCluster.txt", "r");
out=fopen("OutCluster.txt", "w");
fscanf(in, "%s", pbname)
fscanf(in, "%d␣%d␣%d", &M, &N, &ld)
次頁 あ
次頁 い
(1<=ld)&&(ld<=ldmax)
no
printf("Error␣NO␣510")
return 非零の時は必ずこの出力を実行すること
return 510
yes
(2<=M)&&(M<=N)&&(N<=NMAX)
no
return 520
yes
空回避 (K=0; K≦M; K++)
NS[K]←1
空回避
dummy←Prepare-Network(M, N, NS, min, max)
min[K], max[K], NS[K] (1≦K≦M0) の計算, 成功の時のみ0をとる。
dummy : 0
≠
return 530
=
NS[0]←1, NS[M]←1
メモリの Dynamic Allocation (K=0; K<=M; K++)
state␣=new␣int*[M+1];
trace-back␣=new␣int*[M+1];
dpv␣=new␣float*[M+1];
for(K=0; K<=M; K++)
{state[K]= new␣int[NS[K]];
dpv[K]=new␣float[NS[K]];
trace-back=new␣int[NS[K]];
if(! state[K]){
cerr≪"Allocation Error of state["≪K"].520"≪endl; return␣520;}
if(! dpv[K]){
cerr≪"Allocation Error of␣dpv["≪K≪"].530"≪endl; return␣530;}
}
各々を分けて確保せよ。
pointer␣state[K]の後に NS[K]個の int型連続データ域を確保
pointer␣dpv[K]の後に NS[K]個の float型連続データ域を確保
メモリの Dynamic Allocation
A

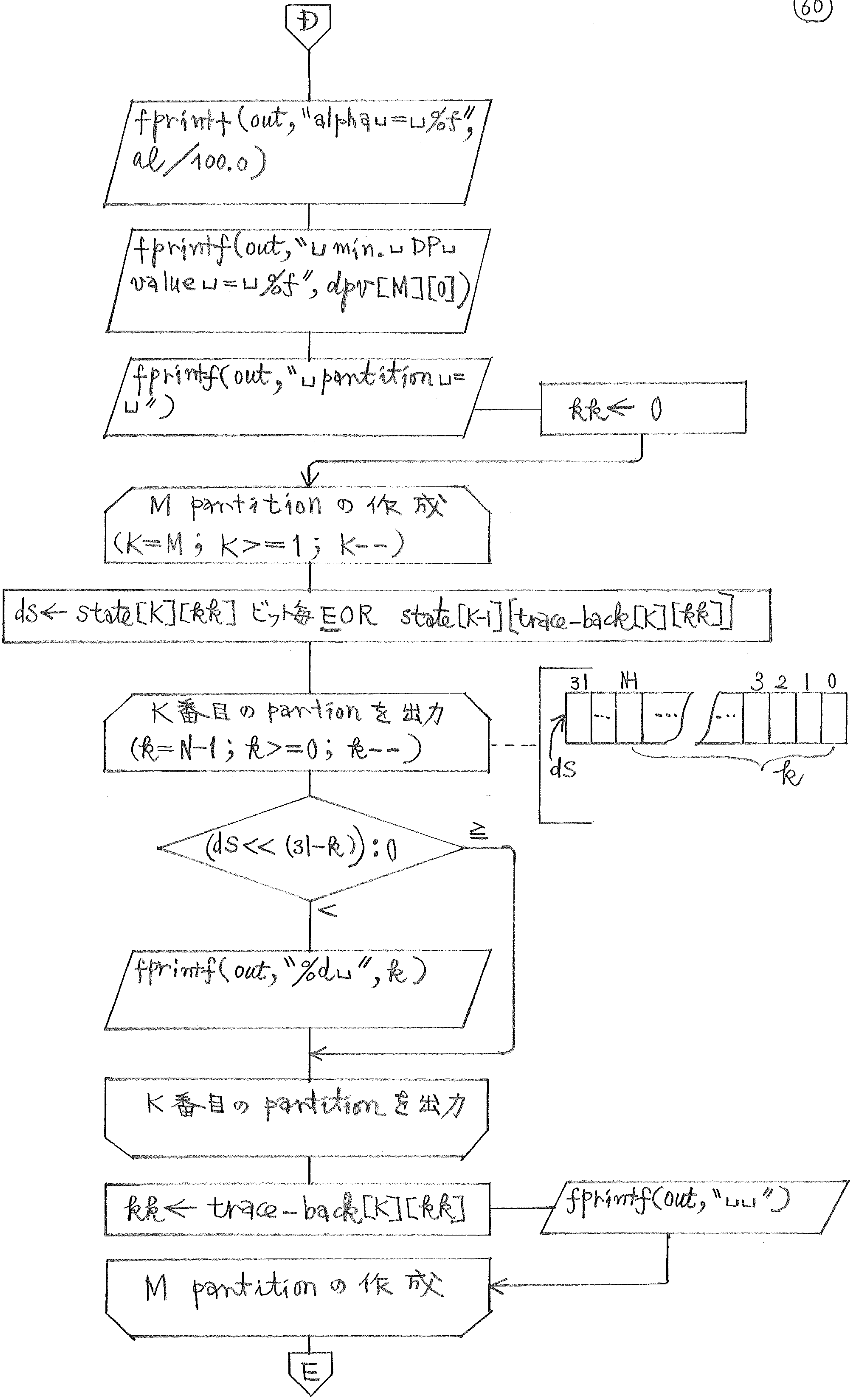

前頁 い
前頁 あ
Nコのデータ
(i=0; i< N ;i++)
fscanf(in, "%d", &dummy)
ld 次元
(k=0; k<ld; k++)
fscanf(in, "%f␣%f", &adn[i][k], &aup[i][k])
ld 次元
Nコのデータ
fprintf(out, "%s¥n", pbname)
fprintf(out, "M␣=␣%d␣N␣=␣%d␣ld␣=␣%d¥n", M, N, ld)
fprintf(out, "¥n¥n")
fprintf(out, "alpha␣␣␣␣␣partition␣␣␣␣␣¥n¥n")

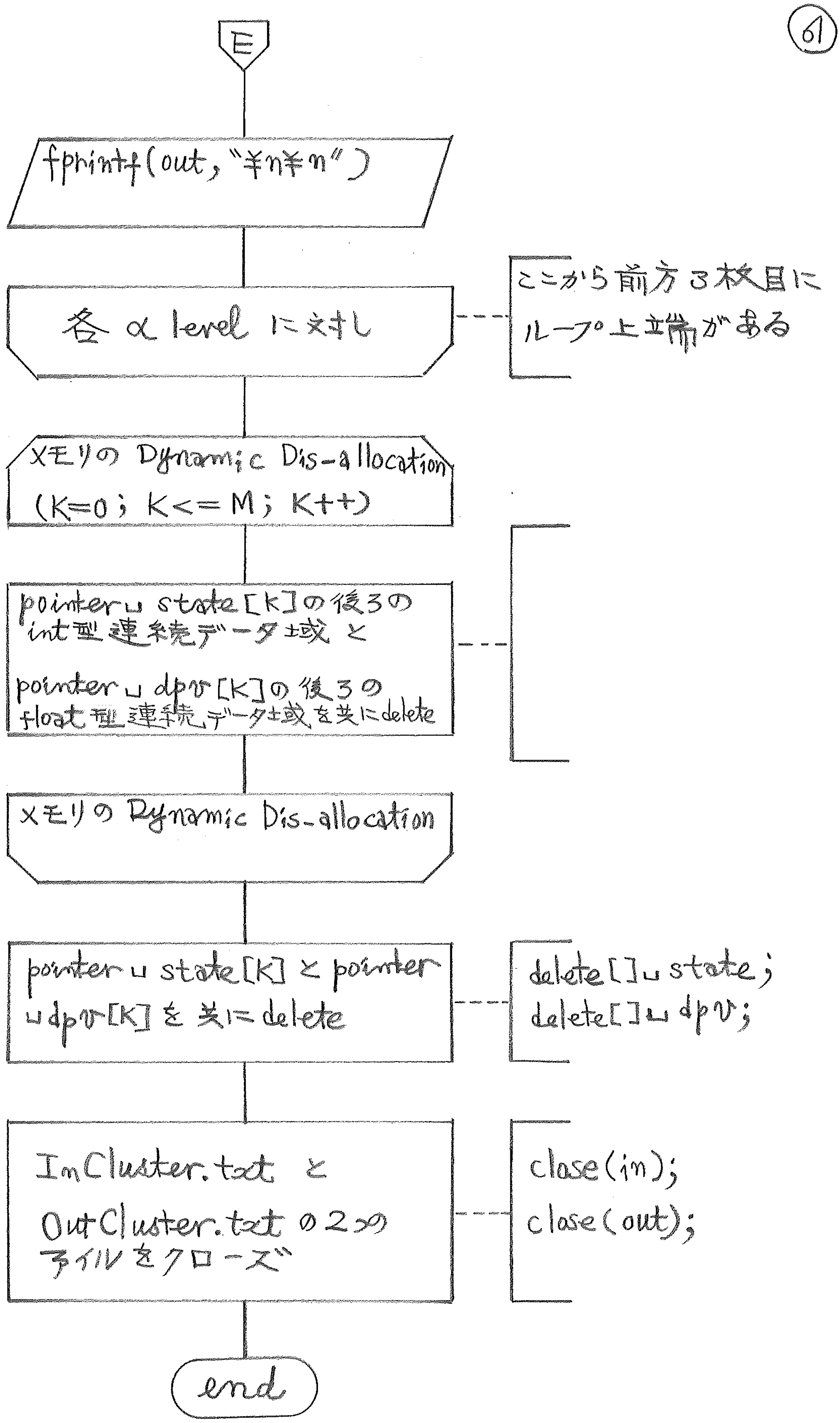

state[K]
ks = 0
1
k=min[K]
元集合,大きさ
(N min[K])
ks
k=min[K]+1
元集合大きさ
(N min[K]+1)
k=min[K]+2
元集合大きさ
(N min[K]+2)
k=max[K]
元集合、
大きさ
(N max[K])
大きさNS[K]の
一次元配列,ここで
NS[K]=Σ_{k=min[K]}^{max[K]} (N k),
1≦K≦M


A
state[0][0]←0
Stage 0 は empty set からなる state 1個のみ.
Stage␣Kの全 states生成
(K=1; K<=M; K++)
ks← 0
k元集合生成
(k=min[K]; k<=max[K]; k++)
Caution! k>=2?
ks←
Subsets(N, k, state[K] , ks)
Stage K の k元 states
の state[K][ ]における
開始番地
k元集合生成
Stage␣Kの全 states生成
B

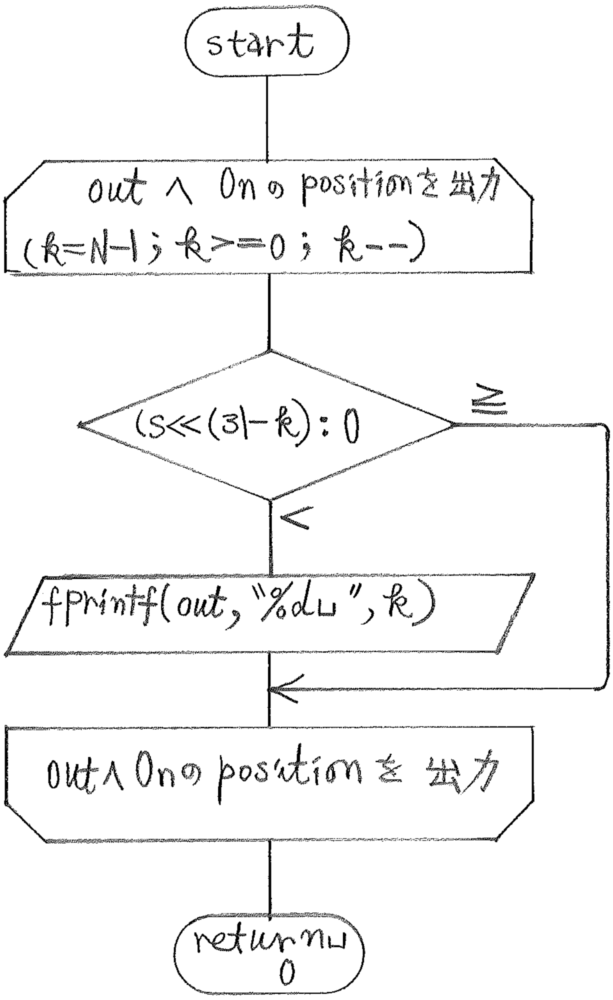

B
dpv[0][0] ← 0.0
Stage 0のempty stateのDP値は0.0.
各α level に対し
(al=1; al<=100; al++)
ここから後へ3枚目にループ下端がある.
dij(α)の計算,
0≦i<j<N.
dij(α)の計算1
(j=1; j<N; j++)
100*N*N*4バイト。 dij(α)
の為に必要なメモリサイズ=400*N*N
バイト。N=25の時, 400*25*25
=250KB。N=32の時
400*32*32=410KB。
dij(α)の計算2
(i=0; i<j; i++)
d[i][j]←d[j][i]←dijal(N, al/100.0,
ld, adn[i], aup[i], adn[j], aup[j])
dij(α)の計算2
dij(α)の計算1
evalのゼロクリア
(kk=0; kk<33554432; kk++)
eval[kk] ← 0.0
evalのゼロクリア
kk ← 0
evalの計算
(k=0; k<=max[1]; k++)
kk ← Makeup-eval(N, k, eval, d, kk)
evalの計算
C


C
K Stage DP 計算
(K=1; K<=M; K++)
differencau set(差集合) dsの作り方
1 0 1 1
排他的OR
1 0 0 0
ds 0 0 1 1
kkの ループ
(kk=0; kk<NS[K] ;kk++)
dpv[K][kk] ← +∞
EOR(Exclusive OR)は排他的ORのこと.
kの ループ
(k=0; k<NS[K-1]; k++)
stateKm1k ← state[K-1][k], is ← stateKm1k と state[K][kk] とのビット毎 &, ds ← state[K][kk] と stateKm1k とのビット毎 EOR
(is==stateKm1k)&&(ds !=0)
no
yes
state[K-1][k] ⊊ state[K][kk]
dpv[K][kk] : dpv[K-1][k]+eval[ds]
≦
>
dpv[K][kk] ← dpv[K-1][k] + eval[ds]
trace_back[K][kk] ← k
Stage K-1
Stage K
0
1
2
0
1
kk
k
ds
kの ループ
kkの ループ
K Stage DP 計算
D


D
fprintf(out, "alpha␣=␣%f", al/100.0)
fprintf(out, "␣min.␣DP␣value␣=␣%f", dpv[M][0])
fprintf(out, "␣partition␣=␣")
kk ← 0
M partition の作成
(K=M; K>=1; K--)
ds ← state[K][kk] ビット毎EOR state[K-1][trace-back[K][kk]]
K番目の partion を出力
(k=N-1; k>=0; k--)
31
N-1
3 2 1 0
ds
k
(ds<<(31-k)):0
≧
<
fprintf(out, "%d␣", k)
K番目の partition を出力
kk ← trace-back[K][kk]
fprintf(out, "␣␣")
M partition の作成
E


E
fprintf(out, "¥n¥n")
各 α level に対し
ここから前方3枚目にループ上端がある
メモリの Dynamic Dis-allocation
(K=0; K<=M; K++)
pointer␣state[K]の後ろのint型連続データ域と
pointer␣dpv[K]の後ろのfloat型連続データ域を共にdelete
メモリの Dynamic Dis-allocation
pointer␣state[K]とpointer␣dpv[K]を共にdelete
delete[]␣state;
delete[]␣dpv;
InCluster.txt と
OutCluster.txt の2つのファイルをクローズ
close(in);
close(out);
end

int型の
1ワードの領域 S 中のビットオンのビットポジションを出力
(Outcluster.txt へ)
する関数 int␣Output_BitonPosition(int␣N, int␣S)



$0 \leqq k \leqq max[1]$なk元集合 sに対する評価値を計算

してeval[s]に代入する。このような計算と代入を各kに対し

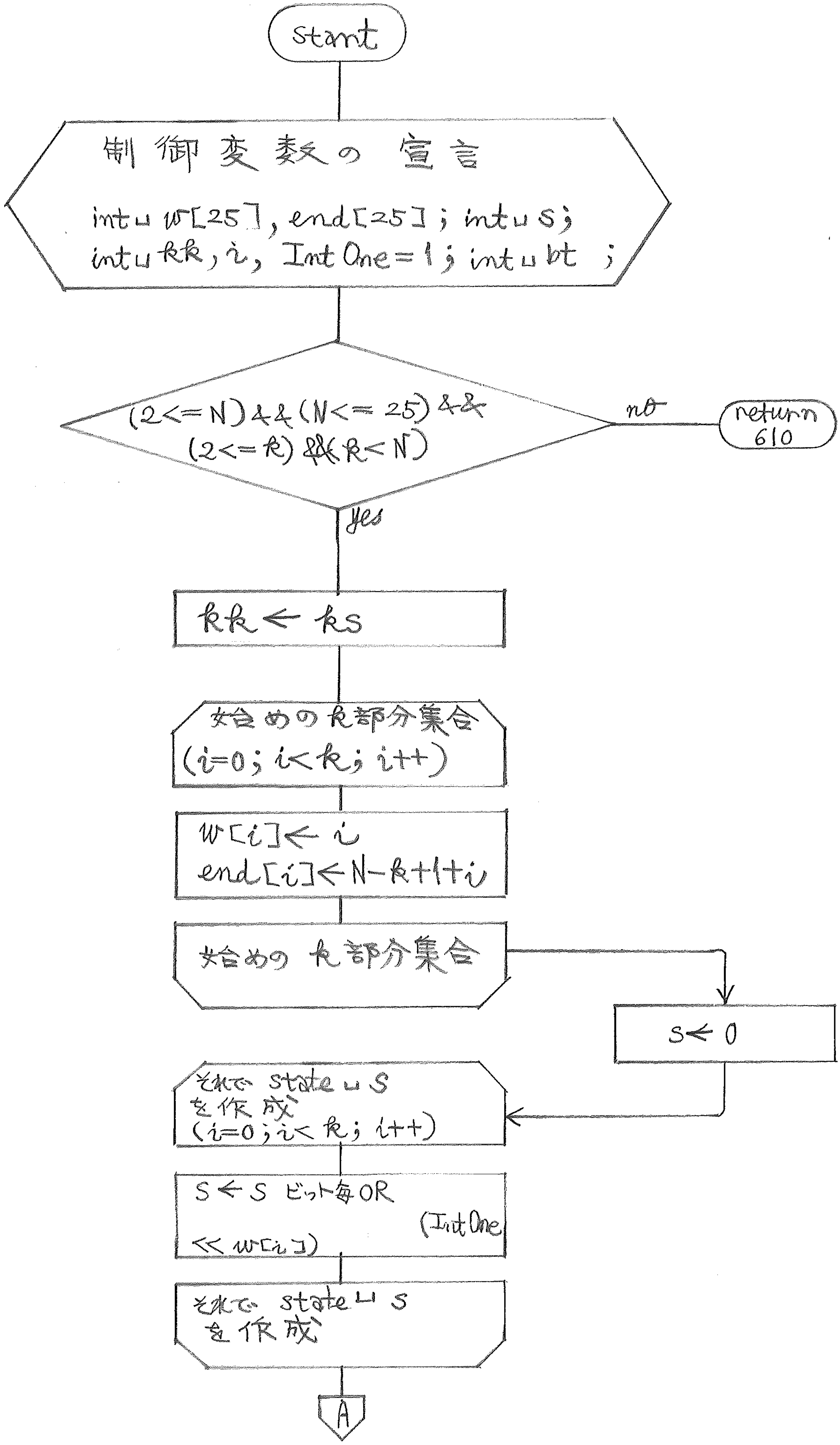
今まで何回行なったかを返す関数 int␣Makeup-eval

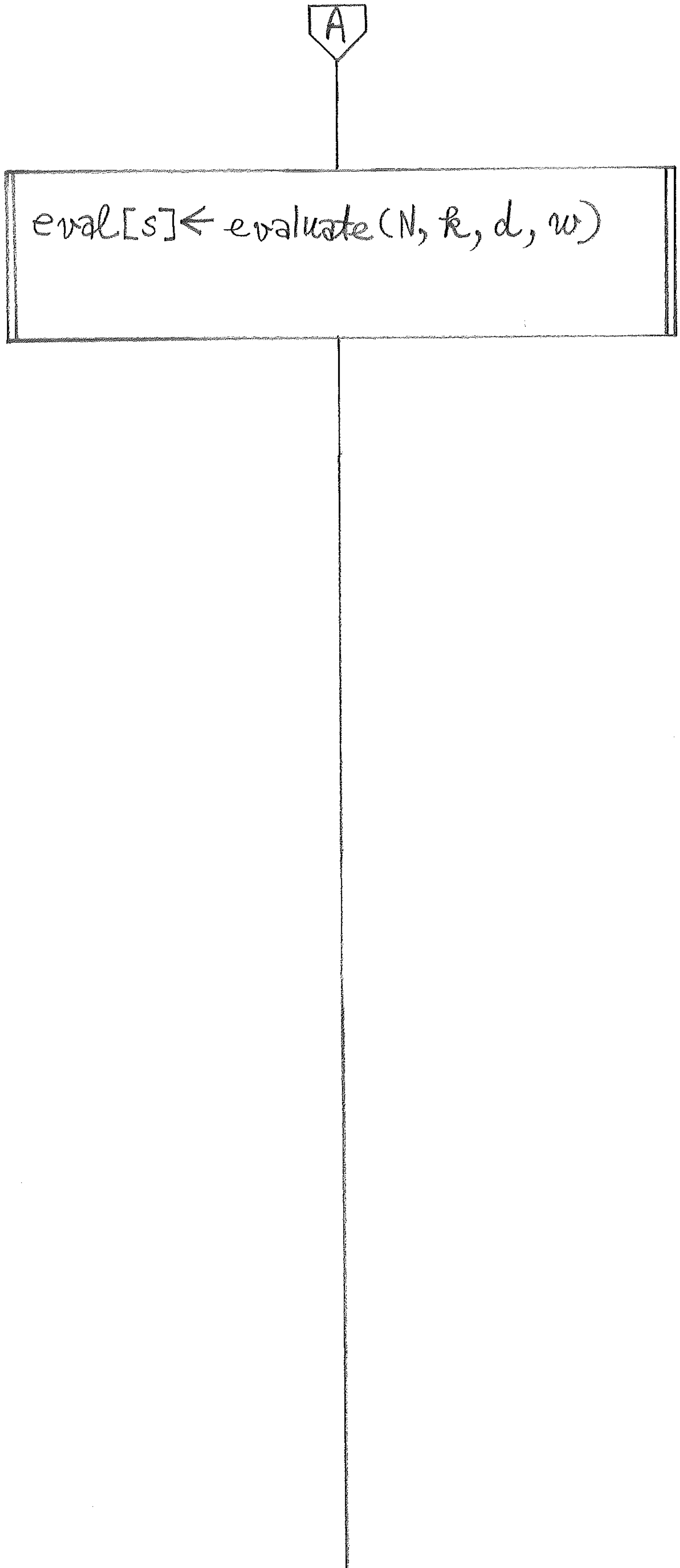
(int␣N, int␣k, float␣eval[], float␣d[][], int␣ks)

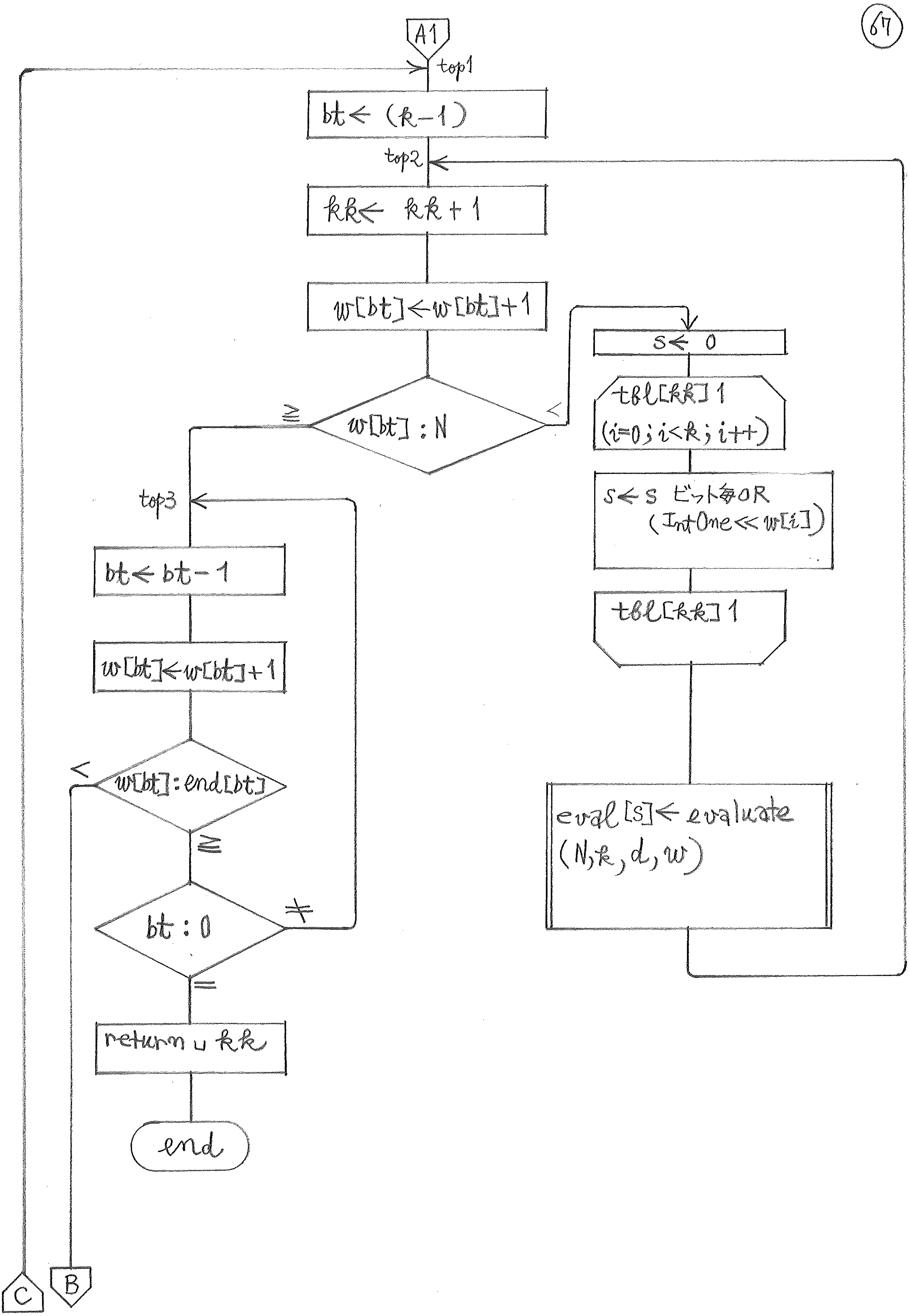
# データ部の定義　int ␣ Makeup_eval

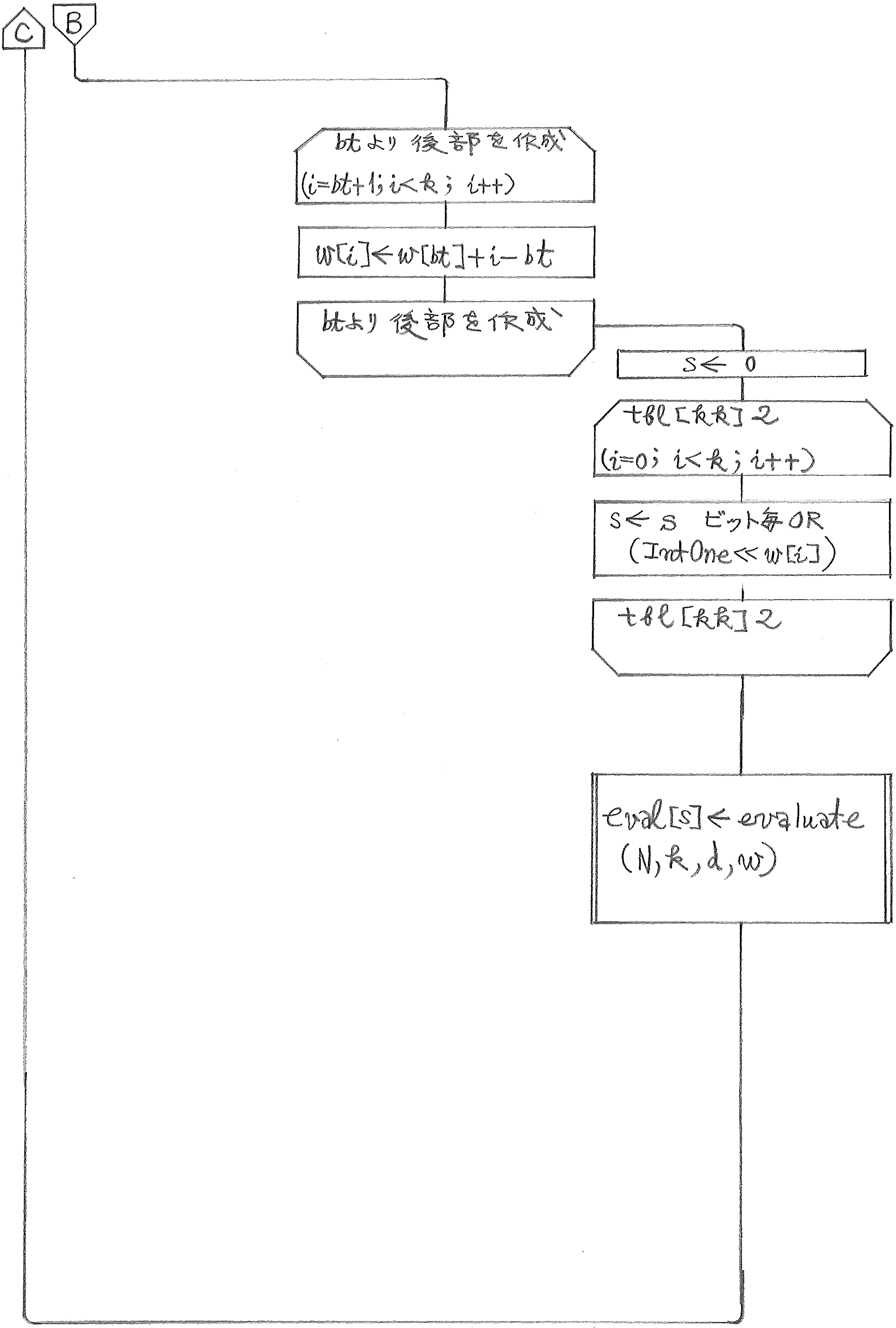
| 変数名または配列名 | データの型，配列の大きさ | 意　味 |
| --- | --- | --- |
| N | int | データ総数，$2 \leqq N \leqq NMAX = 25$. |
| k | int | k元集合のk，$0 \leqq k \leqq max[1]$. |
| eval | float, [ ] | tableau形式で持った評価値 eval，個までの要素からなる集合全てに対して計算する。集合 $W = \{ w[i] \mid 0 \leqq i < k \}$ に対し $eval[W] = \frac{1}{\lvert W \rvert} \sum_{i<j \in W} d_{ij}(\alpha)$. |
| d | float, [ ][ ] | $d[i][j] = d_{ij}(\alpha)$, $0 \leqq i < j < N$. |
| w | int, [NMAX] | $W = \{ w[i] \mid 0 \leqq i < k \}$. |
| s | int | $\{$ sのビット1が立っているビットポジション $\} = W$, $s \overset{1-1}{\leftrightarrow} W$ |
| ks | int | Starting position for k-sets. |
| end | int, [NMAX] | k元集合を作るためのワークエリア $end[i] = N - k + 1 + i \ (0 \leqq i < k)$ |
| kk | int | 0から始まって kk番目に評価する。 |
| i | int | 制御変数 |
| IntOne | int | IntOne = 1. 第0ビットポジションのみビットOn. |
| bt | int | back track用制御変数 |

int␣Makeup_eval(int␣N, int␣k,
float␣eval[ ], float␣d[ ][ ], int␣ks)




A
eval[s] ← evaluate(N, k, d, w)
A1


A1
top1
bt ← (k−1)
top2
kk ← kk+1
w[bt] ← w[bt]+1
w[bt] : N
≧
<
s ← 0
tbl[kk] 1
(i=0; i<k; i++)
s ← s ビット毎OR
(IntOne ≪ w[i])
tbl[kk] 1
eval[s] ← evaluate
(N, k, d, w)
top3
bt ← bt−1
w[bt] ← w[bt]+1
w[bt] : end[bt]
<
≧
bt : 0
≠
=
return ␣ kk
end
C
B

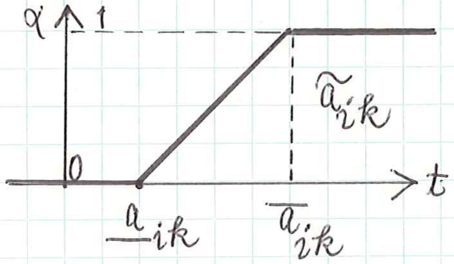

C
B
btより後部を作成
(i=bt+1; i<k; i++)
w[i]←w[bt]+i−bt
btより後部を作成
S← 0
tbl[kk]2
(i=0; i<k; i++)
S← S ビット毎OR
(IntOne≪w[i])
tbl[kk]2
eval[s]←evaluate
(N, k, d, w)

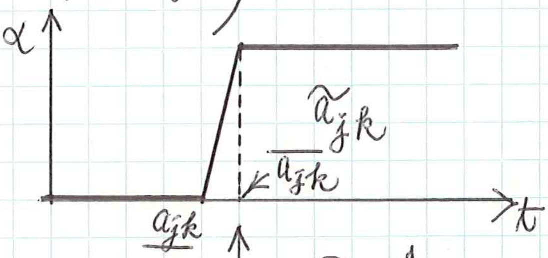
## $\{1, 2, \cdots, N\} \ni i, j\ (i \neq j)$ に対する fuzzy 距離 $d_{ij}(\alpha)$ の計算法, ただし $0 < \alpha \leqq 1$ とする。(蔵野, "Monotone Fuzzy Dataで表わされる Clustering 問題の計算", 2003/11/20による)

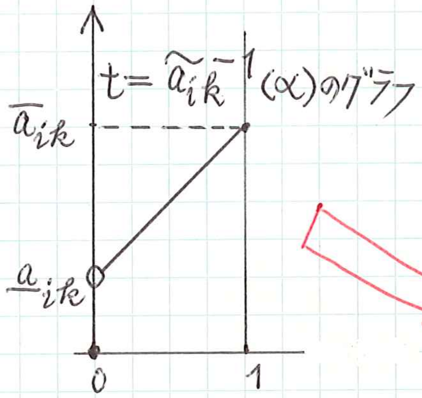
・ $1 \leqq i \leqq N$ に対し $\ell$ 次元 fuzzy data $\tilde{a}_i = (\tilde{a}_{i1}, \tilde{a}_{i2}, \cdots, \tilde{a}_{i\ell}) \in \mathcal{F}_+^{\ell}$ が与えられたとする。ただし $\tilde{a}_{ik} = [\underline{a}_{ik}, \overline{a}_{ik}]$ の形の monotone fuzzy number とする。$1 \leqq k \leqq \ell$。

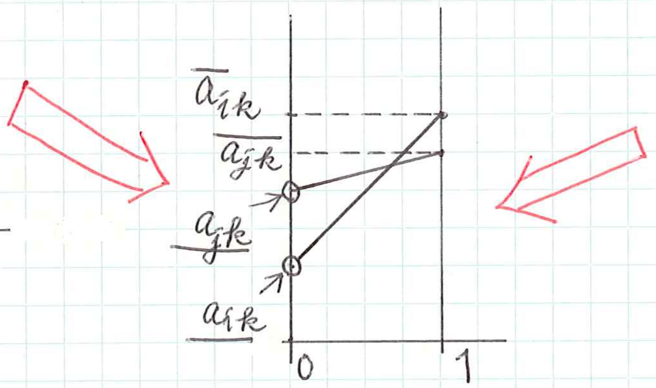
$$\tilde{a}_{ik}^{-1}(\alpha) = \begin{cases} 0, & \alpha = 0 \\ (\overline{a}_{ik} - \underline{a}_{ik})\alpha + \underline{a}_{ik}, & 0 < \alpha \leqq 1. \end{cases}$$

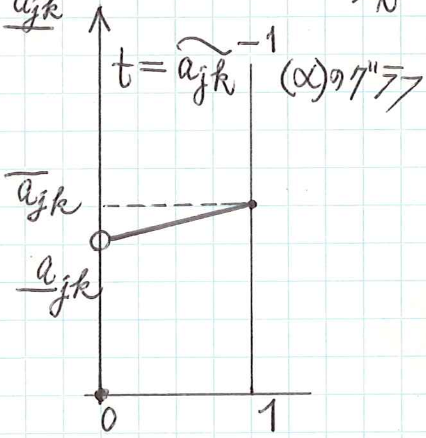
$$\left|\tilde{a}_{ik}^{-1}(\alpha) - \tilde{a}_{jk}^{-1}(\alpha)\right| = \begin{cases} 0, & \alpha = 0 \\ \left|\left((\underline{a}_{jk} - \underline{a}_{ik}) + (\overline{a}_{ik} - \overline{a}_{jk})\right)\alpha + \underline{a}_{ik} - \underline{a}_{jk}\right|, & 0 < \alpha \leqq 1. \end{cases}$$




・ $$f(\alpha) = \sum_{k=1}^{\ell}\left(\tilde{a}_{ik}^{-1}(\alpha) - \tilde{a}_{jk}^{-1}(\alpha)\right)^2$$
$$= \sum_{k=1}^{\ell}\left[\left((\underline{a}_{jk} - \underline{a}_{ik}) + (\overline{a}_{ik} - \overline{a}_{jk})\right)\alpha + \underline{a}_{ik} - \underline{a}_{jk}\right]^2$$

ここで $p_k = (\underline{a}_{jk} - \underline{a}_{ik}) + (\overline{a}_{ik} - \overline{a}_{jk})$ とおくと, $0 < \alpha \leqq 1$ に対し

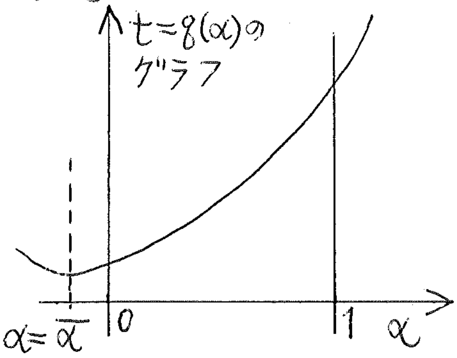
$$f(\alpha) = \sum_{k=1}^{\ell}\left[p_k\alpha + \underline{a}_{ik} - \underline{a}_{jk}\right]^2$$
$$= \left(\sum_{k=1}^{\ell} p_k^2\right)\alpha^2 + 2\sum_{k=1}^{\ell} p_k(\underline{a}_{ik} - \underline{a}_{jk})\alpha + \sum_{k=1}^{\ell}(\underline{a}_{ik} - \underline{a}_{jk})^2$$

$Q(\alpha) = \sup_{0 \leqq \alpha' \leqq \alpha} f(\alpha')$, $\rho(\alpha) \equiv \rho(\alpha \mid \tilde{a}_i, \tilde{a}_j) \equiv \sqrt{Q(\alpha)}$ とおく。

- $0<\alpha\leqq 1$ での $\alpha$ の2次関数 $t=g(\alpha)$ の様子と $Q(\alpha)$ の決定．

$$\bar{\alpha} \triangleq \frac{-\sum_{k=1}^{\ell} p_k(a_{ik}-a_{jk})}{\sum_{k=1}^{\ell} p_k^2}, \quad \sum_{k=1}^{\ell} p_k^2 > 0 \text{ とする。}$$

$\bar{\alpha} \leqq 0 \left(\sum_{k=1}^{\ell} p_k(a_{ik}-a_{jk})\right)$ のとき

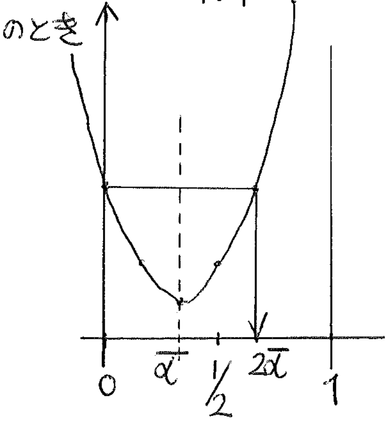



グラフより

$Q(\alpha) \triangleq g(\alpha)$

とおけばよい。

$0<\bar{\alpha}<\frac{1}{2}$ $\left(-\frac{\sum p_k^2}{2} < \sum_{k=1}^{\ell} p_k(a_{ik}-a_{jk}) < 0\right)$ のとき



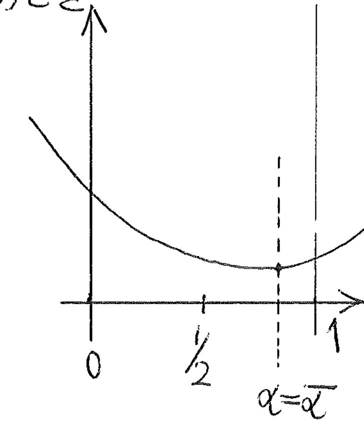

グラフより

$0<\alpha\leqq 2\bar{\alpha}$ で $Q(\alpha)=g(0)=\sum_{k=1}^{\ell}(a_{ik}-a_{jk})^2$,

$2\bar{\alpha}<\alpha\leqq 1$ で $Q(\alpha)=g(\alpha)$

とおけばよい。

$\frac{1}{2}\leqq\bar{\alpha}$ $\left(\sum_{k=1}^{\ell} p_k(a_{ik}-a_{jk})^2 \leqq -\frac{1}{2}\sum_{k=1}^{\ell} p_k^2\right)$ のとき




グラフより

$Q(\alpha)=g(0)=\sum_{k=1}^{\ell}(a_{ik}-a_{jk})^2$

とおけばよい。

- $d_{ij}(\alpha) \triangleq p(\alpha) \equiv \sqrt{Q(\alpha)} \triangleq p(\alpha \mid \tilde{a}_i, \tilde{a}_j)$

- int 変数 al を用いる。

```
for(al=1; al<=100; al++)
    {レベル値 α= al/100.0 に対し
        ã_i, ã_j の距離 ← d_ij(α)

        for(1≦i<j≦N)
    }
```

とすればよい。

- $\{1,2,\cdots,N\}$ を $\{0,1,\cdots,(N-1)\}$ で，$1\leqq i\leqq N$ を $0\leqq i\leqq(N-1)$ で，$\sum_{k=1}^{\ell}$ を $\sum_{k=0}^{(\ell-1)}$ と変更してCプログラムを作成すること。

・実質入力データは float 型の

$N*l*2$ 個

である。これを保存するための配列の大きさは

$8*N*l$ バイト

である。

$N=25,\ l=10$ で　$8*25*10=2K$ byte.

$N=25,\ l=20$ で　$8*25*20=4K$ byte.

・$d_{ij}(\alpha)$ を保持するために要するメモリサイズは

($\alpha$のきざみ幅)$*N*N*4$ バイトである。Memory の Dynamic Allocation を利用して ($\alpha$のきざみ幅)$*\frac{N*(N+1)}{2}*4$ バイトまで減らせるが、減らす努力が必要か？ N の最大値は NMAX=25 だから

$\alpha$のきざみ幅＝100の時

$100*25*25*4=250KB$

$\alpha$のきざみ幅＝1000の時

$1000*25*25*4=2.5MB$.

努力するかいはありそうだ。

・各$\alpha$毎に$d_{ij}(\alpha)$を計算する。従って$\alpha$毎に2次元配列$d_{ij}$を作り直すことにすれば必要バイト数は$\alpha$のきざみ幅に依存せず $N*N*4$ バイト必要である。

$N=25$の時　$N*N*4=25*25*4=25*100=2500$バイト$\fallingdotseq 3K$バイト.

・$\{1,2,\cdots,N\}\supseteq A$ に対し A の評価値 $eval(A)=\frac{1}{|A|}\sum_{\substack{i,j\in A\\ i<j}} d_{ij}(\alpha)$

で計算する。

各$\alpha$ level値に対する$d_{ij}(\alpha)$を計算する

関数 float␣dijal(int␣N,␣float␣alpha,␣int␣ld,

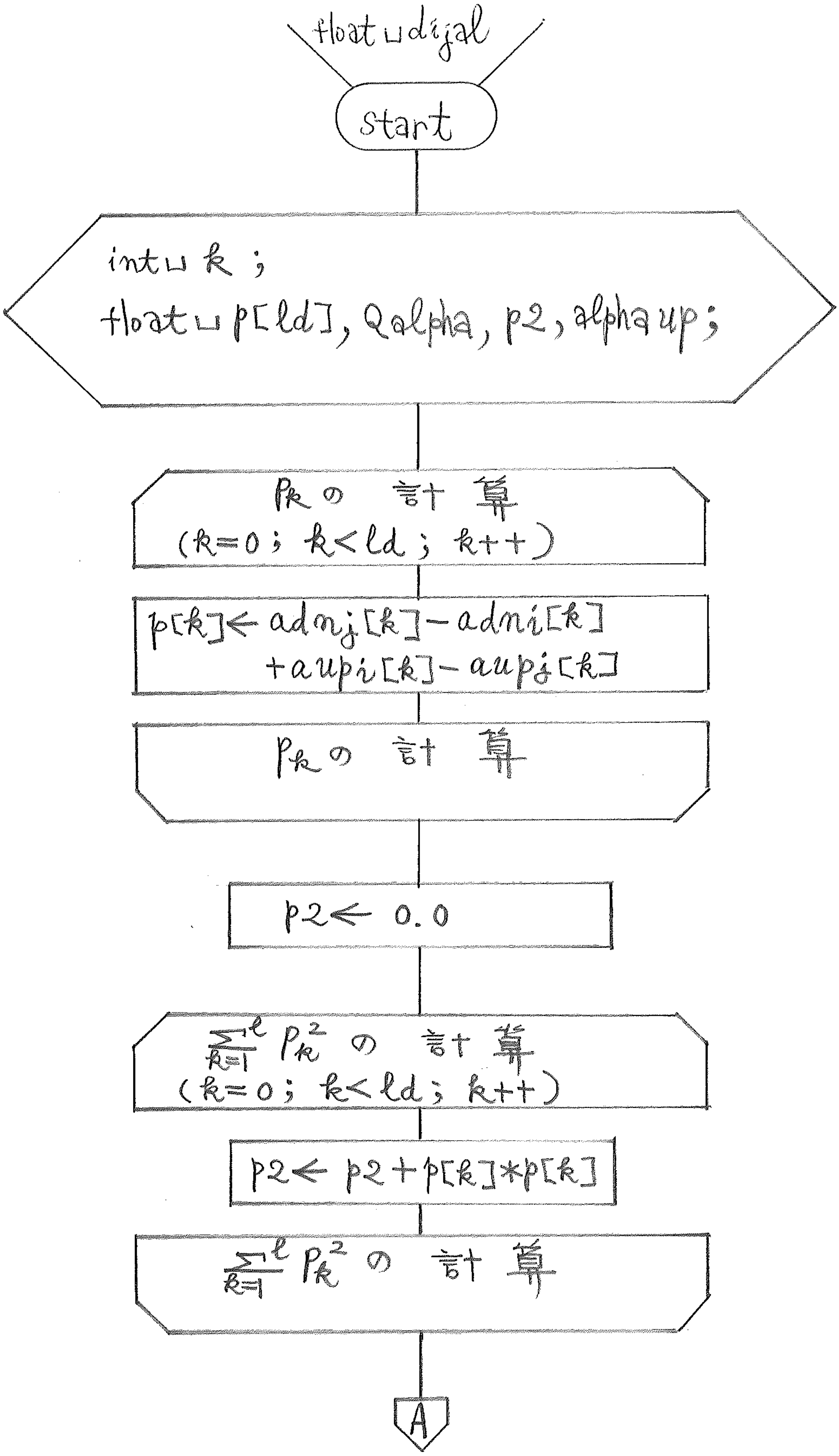
␣float␣*adni,␣float␣*aupi,␣float␣adnj[],␣float␣aupj[])

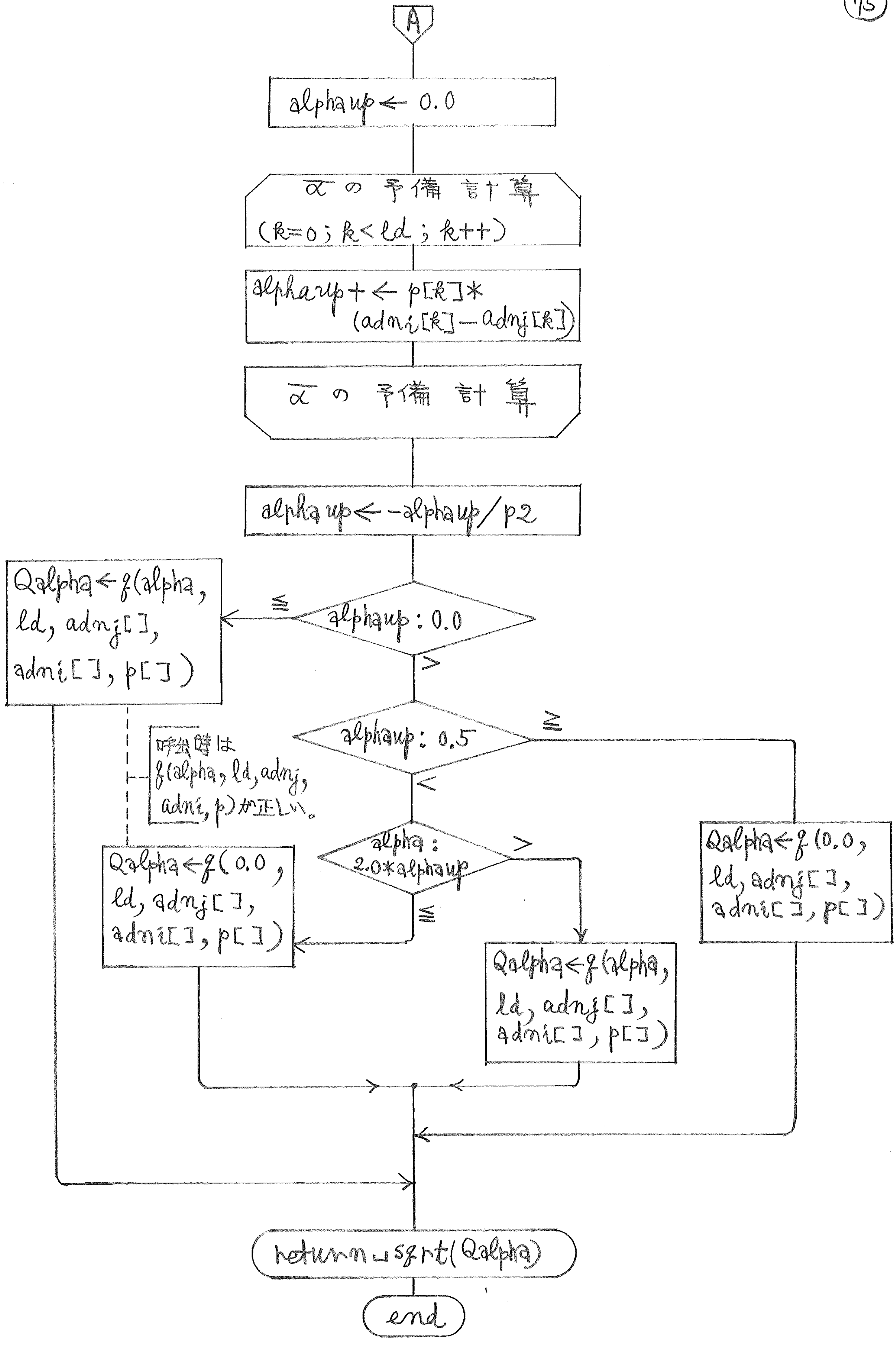
float␣dijal(int␣N, float␣alpha, ␣int␣ld, ␣float␣*adni, ␣float␣*aupi, ␣float␣adnj[], ␣float␣aupj[])

# データ部の定義 関数



| 変数名または配列名 | データの型，配列の大きさ | 意味 |
|---|---|---|
| N | int | データ総数，2≦N≦25，入力データ |
| alpha | float | レベル値$\alpha$，$0<\alpha\leqq 1$。 |
| ld | int | 各データの次元，ld 次元とする。$1\leqq ld\leqq ldmax$。 |
| ldmax | int, global | ld の最大値，$1\leqq ld\leqq ldmax$。 |
| k ✓ | int | 制御変数，$1\leqq k\leqq l$。 |
| adni | float, [ldmax] | adn[i][k] = $\underline{a}_{(i+1)(k+1)}$，$0\leqq i<N$ and $0\leqq k<ld$。<br>// a down から付けた名前。 |
| aupi | float, [ldmax] | aup[i][k] = $\overline{a}_{(i+1)(k+1)}$，$0\leqq i<N$ and $0\leqq k<ld$。<br>// a up から付けた名前。 |
| p ✓ | float [ldmax] | $p_k=(\underline{a}_{jk}-\underline{a}_{ik})+(\overline{a}_{ik}-\overline{a}_{jk})$ |
| Qalpha ✓ | float | $Q(\alpha)=Q($alpha$)$ |
| p2 ✓ | float | $p2=\sum_{k=1}^{l} p_k^2$ |
| alphaup ✓ | float | $\overline{\alpha}=\frac{-\sum_{k=1}^{l} p_k(\underline{a}_{ik}-\underline{a}_{jk})}{\sum_{k=1}^{l} p_k^2}=\frac{-\sum_{k=1}^{l} p_k(\underline{a}_{ik}-\underline{a}_{jk})}{p2}$ |
| adnj | float, [ldmax] | adn[j][k] = $\underline{a}_{(j+1)(k+1)}$ が入る。$0\leqq j<N$ and $0\leqq k<ld$。<br>// a␣down |
| aupj | float [ldmax] | aup[j][k] = $\overline{a}_{(j+1)(k+1)}$ が入る。$0\leqq j<N$ and $0\leqq k<ld$。<br>// a␣up |


float dijal
start
int k;
float p[ld], Qalpha, p2, alphaup;
Pkの 計 算
(k=0; k<ld; k++)
p[k]← adnj[k]−adni[k]
+aupi[k]−aupj[k]
Pkの 計 算
p2← 0.0
Σ_{k=1}^{l} Pk² の 計 算
(k=0; k<ld; k++)
p2← p2+p[k]*p[k]
Σ_{k=1}^{l} Pk² の 計 算
A

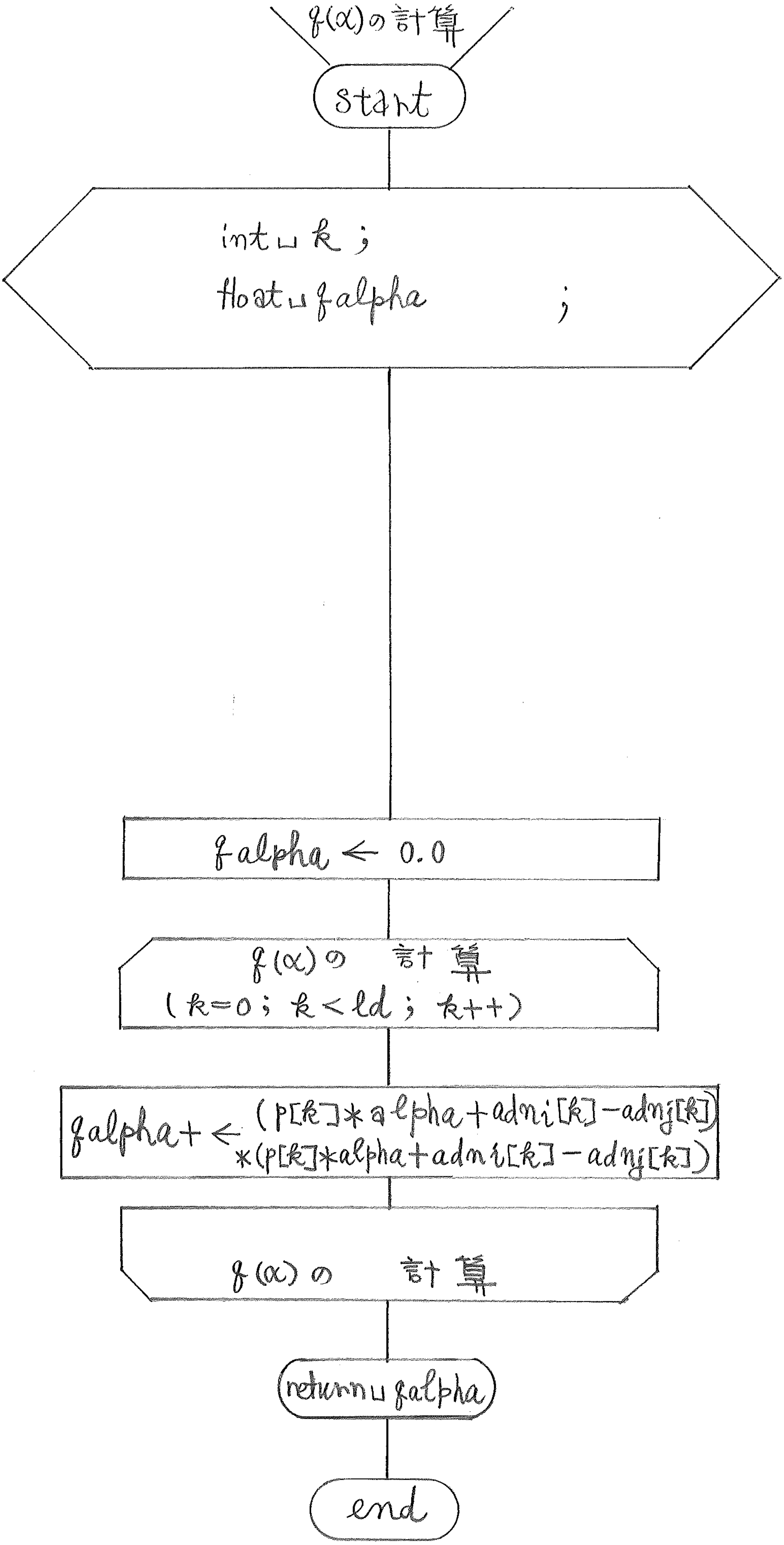

A
alphaup ← 0.0
α の 予備 計算
(k=0; k<ld; k++)
alphaup+ ← p[k]*
(adni[k]−adnj[k])
α の 予備 計算
alphaup ← −alphaup/p2
alphaup : 0.0
≦
>
Qalpha ← g(alpha,
ld, adnj[ ],
adni[ ], p[ ])
alphaup : 0.5
≧
<
呼出時は
g(alpha, ld, adnj,
adni, p)が正しい。
alpha :
2.0*alphaup
>
≦
Qalpha ← g(0.0,
ld, adnj[ ],
adni[ ], p[ ])
Qalpha ← g(alpha,
ld, adnj[ ],
adni[ ], p[ ])
Qalpha ← g(0.0,
ld, adnj[ ],
adni[ ], p[ ])
return␣sqrt(Qalpha)
end

関数 $f(x)$ の計算　float␣f(␣float␣alpha, int␣ld, float␣adnj[],

float␣adni[], float␣p[]　　　　　　　)

# データ部の定義　float␣f

| 変数または配列名 | 型，大きさ | 意味 |
|---|---|---|
| alpha | float | $alpha = al/100.0$，$1 \leqq al \leqq 100$ |
| ld | int | データの次元，$1 \leqq ld \leqq ldmax$。 |
| adnj | float, [ld] | $adnj[k] = \underline{a_{jk}}$，$1 \leqq k \leqq ld$. |
| adni | float, [ld] | $adni[k] = \underline{a_{ik}}$，$1 \leqq k \leqq ld$. |
| aupi | float, [ld] | $aupi[k] = \overline{a_{ik}}$，$1 \leqq k \leqq ld$，使用中止 |
| aupj | float, [ld] | $aupj[k] = \overline{a_{jk}}$，$1 \leqq k \leqq ld$，使用中止 |
| k | int | 制御変数 |
| qalpha | float | $q(\alpha) = \sum_{k=1}^{l} [-\underline{a_{jk}} + \underline{a_{ik}} + p_k \alpha \quad ]^2$ |
| p | float, [ld] | $p_k = \underline{a_{jk}} - \underline{a_{ik}} + \overline{a_{ik}} - \overline{a_{jk}}$，$1 \leqq k \leqq ld$. |



f(α)の計算
start
int␣k ;
float␣falpha ;
falpha ← 0.0
f(α)の 計 算
( k=0 ; k<ld ; k++)
falpha+ ← (p[k]*alpha+adni[k]−adnj[k])
*(p[k]*alpha+adni[k]−adnj[k])
f(α)の 計 算
return␣falpha
end

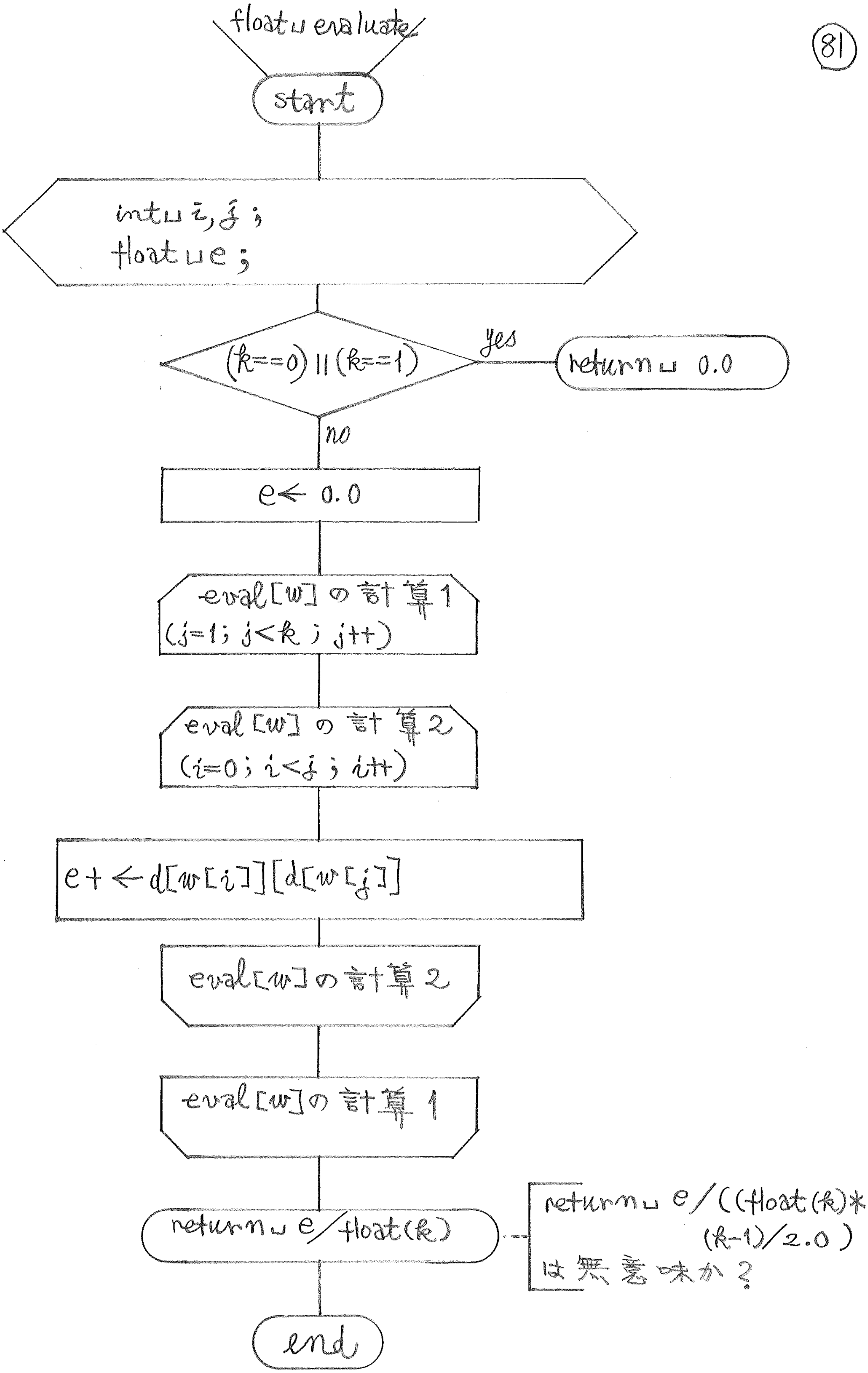
与えられた $k$ 元 state $W$ に対し $\mathrm{eval}(W) = \frac{1}{|W|} \sum_{\substack{i,j \in W \\ i<j}} d_{ij}(x)$

を計算する関数 float␣evaluate(int␣N, int␣k,

float␣d[][], int␣w[])

データ部の定義　　float ⊔ evaluate

| 変数及び配列名 | 型，大きさ | 意味 |
|---|---|---|
| N | int | データ総数，$2 \leqq N \leqq NMAX = 25$. |
| k | int | 集合 $W = w$ の大きさ，$0 \leqq k \leqq N(1)$. |
| i, j | int | 制御変数 |
| d | float, [ ][ ] | $d_{ij}(\alpha) = d[i][j]$ |
| w | int, [ ] | $\{w[i] \mid 0 \leqq i < k\} = W$ |
| e | float | $k=0$ 又は 1 で $e = 0.0$, $k \geqq 2$ で $e = eval(W) = \frac{1}{\lvert W \rvert} \sum_{\substack{i,j \in W \\ i<j}} d_{ij}(\alpha)$ |


float␣evaluate
start
int␣i,j;
float␣e;
(k==0)||(k==1)
yes
return␣0.0
no
e← 0.0
eval[w]の計算1
(j=1; j<k ; j++)
eval[w]の計算2
(i=0; i<j; i++)
e+←d[w[i]][d[w[j]]
eval[w]の計算2
eval[w]の計算1
return␣e/float(k)
return␣e/((float(k)*
(k-1)/2.0)
は無意味か？
end

# デバック用テストデータ ：2004.1.16作成（その1）

$N=5,\ M=3,\ l=ld=2.$ $\{0,1,2,3,4\}$を3つに分割する。$M_0\equiv M=3$とする。

| $i$ \ $l$ | 1(0) | 2(1) |
|---|---|---|
| 1(0) | [0,1] | [1,2] |
| 2(1) | [0,2] | [1,3] |
| 3(2) | [0,3] | [1,4] |
| 4(3) | [0,4] | [2,3] |
| 5(4) | [2,4] | [3,4] |

$max(1)=5-3+1=3$

$max(2)=5-3+2=4$

$max(3)=5-3+3=5$

$N-M[N/M]=5-3\cdot 1=2$

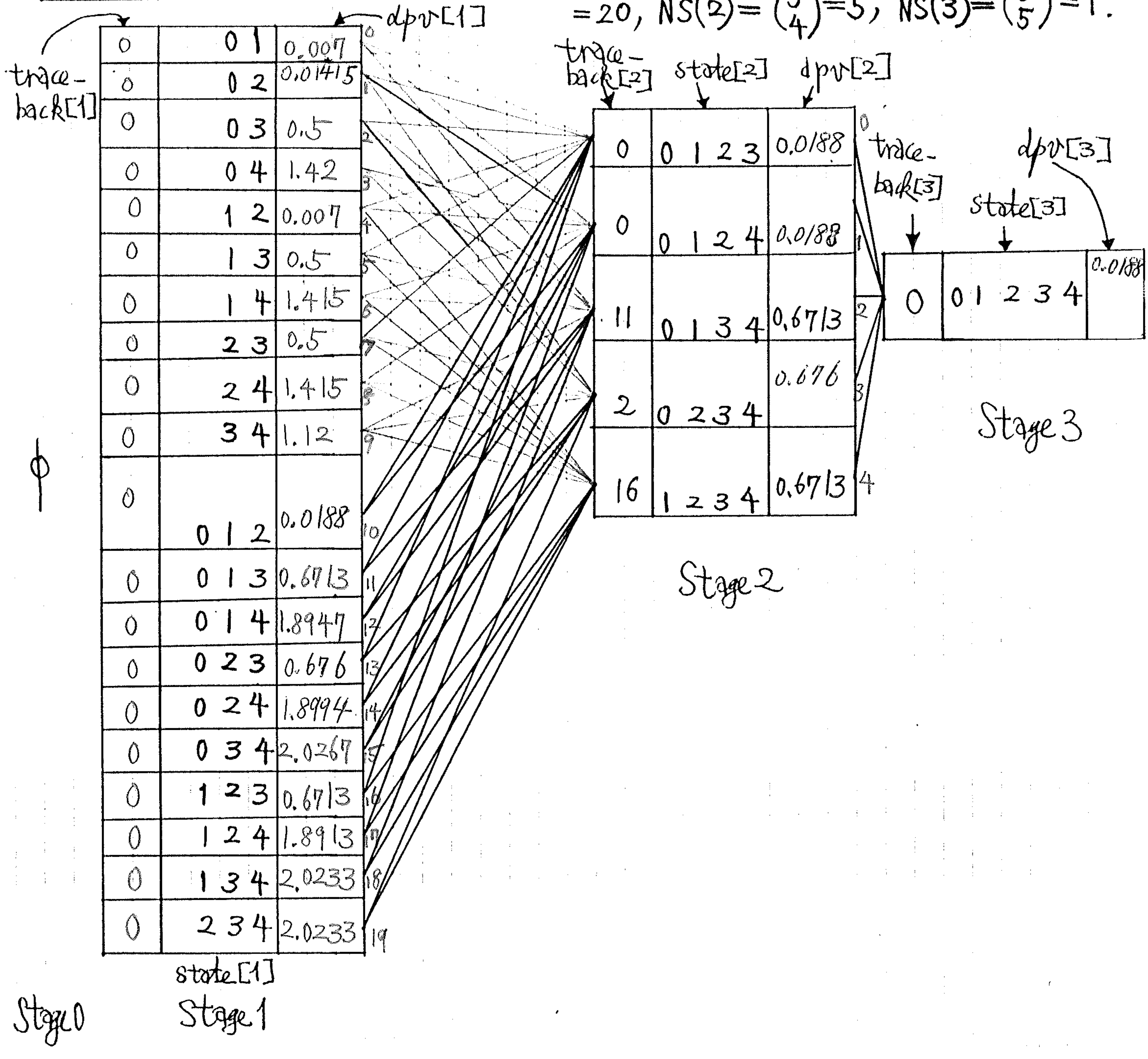
$min(1)=(1+1)\cdot 1=2$

$min(2)=(1+1)\cdot 2=4$

$min(3)=5-(3-3)\cdot 1=5$

$NS(0)=1,\ NS(1)=\sum_{l=2}^{3}\binom{5}{l}=\binom{5}{2}+\binom{5}{2}=2\cdot\frac{5\cdot 4}{2}$

$=20,\ NS(2)=\binom{5}{4}=5,\ NS(3)=\binom{5}{5}=1.$



2004.1.16作成（その2），$\alpha=0.01$.



ビット・ポジション

| | 0 | 1 | 2 | 3 | 4 | eval | |
|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0.0 | |
| 1 | 1 | 0 | 0 | 0 | 0 | 0.0 | |
| 2 | 0 | 1 | 0 | 0 | 0 | 0.0 | |
| 3 | 1 | 1 | 0 | 0 | 0 | 0.007 | $\frac{1}{2}d_{01}(0.001)=0.007$ |
| 4 | 0 | 0 | 1 | 0 | 0 | 0.0 | |
| 5 | 1 | 0 | 1 | 0 | 0 | ~~1.01~~ 0.01415 | $\frac{1}{2}d_{02}(0.01)=1.01$ |
| 6 | 0 | 1 | 1 | 0 | 0 | 0.007 | $\frac{1}{2}d_{12}=0.007$ |
| 7 | 1 | 1 | 1 | 0 | 0 | ~~0.6827~~ 0.0188 | $\frac{1}{3}(d_{01}+d_{02}+d_{12})=(0.014+\cancel{2.02}\,0.0283+0.014)/3=\cancel{2.048}\,0.0563/3=\cancel{0.683}\,0.0188$ |
| 8 | 0 | 0 | 0 | 1 | 0 | 0.0 | |
| 9 | 1 | 0 | 0 | 1 | 0 | 0.5 | $\frac{1}{2}(d_{03})=0.5$ |
| 10 | 0 | 1 | 0 | 1 | 0 | 0.5 | $\frac{1}{2}d_{13}=0.5$ |
| 11 | 1 | 1 | 0 | 1 | 0 | 0.6713 | $(d_{01}+d_{03}+d_{13})/3=(0.014+1.0+1.0)/3=2.014/3=0.6713$ |
| 12 | 0 | 0 | 1 | 1 | 0 | 0.5 | $d_{23}/2=0.5$ |
| 13 | 1 | 0 | 1 | 1 | 0 | ~~1.34~~ 0.676 | $(d_{02}+d_{03}+d_{23})/3=(\cancel{2.02}\,0.0283+1.0+1.0)=\cancel{4.02}\,2.0283/3=\cancel{1.34}\,0.676$ |
| 14 | 0 | 1 | 1 | 1 | 0 | 0.6713 | $(d_{12}+d_{13}+d_{23})/3=(0.014+1.0+1.0)=2.014/3=0.671$ |
| 15 | 1 | 1 | 1 | 1 | 0 | | |
| 16 | 0 | 0 | 0 | 0 | 1 | 0.0 | |
| 17 | 1 | 0 | 0 | 0 | 1 | 1.42 | $d_{04}/2=1.42$ |
| 18 | 0 | 1 | 0 | 0 | 1 | 1.415 | $d_{14}/2=1.415$ |
| 19 | 1 | 1 | 0 | 0 | 1 | 1.8947 | $(d_{01}+d_{04}+d_{14})/3=(0.014+2.84+2.83)/3=5.684/3=1.8947$ |
| 20 | 0 | 0 | 1 | 0 | 1 | 1.415 | $d_{24}/2=1.415$ |
| 21 | 1 | 0 | 1 | 0 | 1 | ~~2.5633~~ 1.8994 | $(d_{02}+d_{04}+d_{24})/3=(\cancel{2.02}\,0.0283+2.84+2.83)/3=\cancel{7.69}\,5.6883/3=\cancel{2.5633}\,1.8994$ |
| 22 | 0 | 1 | 1 | 0 | 1 | 1.8913 | $(d_{12}+d_{14}+d_{24})/3=(0.014+2.83+2.83)/3=5.674/3=1.8913$ |
| 23 | 1 | 1 | 1 | 0 | 1 | | |
| 24 | 0 | 0 | 0 | 1 | 1 | 1.12 | $d_{34}/2=1.12$ |
| 25 | 1 | 0 | 0 | 1 | 1 | 2.0267 | $(d_{03}+d_{04}+d_{34})/3=(1.0+2.84+2.24)/3=6.08/3=2.0267$ |
| 26 | 0 | 1 | 0 | 1 | 1 | 2.0233 | $(d_{13}+d_{14}+d_{34})/3=(1.0+2.83+2.24)/3=6.07/3=2.0233$ |
| 27 | 1 | 1 | 0 | 1 | 1 | | |
| 28 | 0 | 0 | 1 | 1 | 1 | 2.0233 | $(d_{23}+d_{24}+d_{34})/3=(1.0+2.83+2.24)/3=6.07/3=2.0233$ |
| 29 | 1 | 0 | 1 | 1 | 1 | | |
| 30 | 0 | 1 | 1 | 1 | 1 | | |
| 31 | 1 | 1 | 1 | 1 | 1 | | |

ステイト　eval

$al = 1,\ \alpha = 0.01$ の時の $d_{ij}(\alpha)$ の値, $ld = 2 = l$.

$d_{01}(\alpha) = ?$

$p_k \equiv \underline{a}_{jk} - \underline{a}_{ik} + (\overline{a}_{ik} - \overline{a}_{jk}),\ 0 \leqq k < l,\ 0 \leqq i < j < N.$

$\overline{\alpha} \equiv - \sum_k p_k(\underline{a}_{ik} - \underline{a}_{jk}) / \sum_k p_k^2.$

$i\ j$

$k=0$: $p_0 = (\underline{a}_{10} - \underline{a}_{00}) + (\overline{a}_{00} - \overline{a}_{10}),\ \underline{a}_{00} - \underline{a}_{10} = 0 - 0 = 0,$

$= 0 - 0 + 1 - 2 = -1$

$k=1$: $p_1 = \underline{a}_{11} - \underline{a}_{01} + \overline{a}_{01} - \overline{a}_{11},\ \underline{a}_{01} - \underline{a}_{11} = 1 - 1 = 0,$

$= 1 - 1 + 2 - 3 = -1$

$\sum_k p_k(\underline{a}_{ik} - \underline{a}_{jk}) = 0,$

$\overline{\alpha} = \frac{-0}{\sum_{k=0}^{1} p_k^2} = \frac{-0}{1+1} = 0 \leqq 0$

$g(0) = \sum_{k=0}^{1}(\underline{a}_{0k} - \underline{a}_{1k})^2 = (0-0)^2 + (1-1)^2 = 0,$

$g(\alpha) = 2\alpha^2 + 0\cdot\alpha + 0 = 2\alpha^2 = 2\left(\frac{1}{10000}\right) = \frac{2}{10000} = 0.0002$

$d_{01}(\alpha) = \sqrt{Q(\alpha)} = \sqrt{g(\alpha)} = \sqrt{2}\alpha = \frac{\sqrt{2}}{100} = \frac{1.414}{100} = 0.01414$

$d_{02}(\alpha) = ?$

$p_0 = (\underline{a}_{20} - \underline{a}_{00}) + (\overline{a}_{00} - \overline{a}_{20}),\ \underline{a}_{00} - \underline{a}_{20} = 0 - 0 = 0,$

$= 0 - 0 + 1 - 3 = -2$

$p_1 = (\underline{a}_{21} - \underline{a}_{01}) + (\overline{a}_{01} - \overline{a}_{21}),\ \underline{a}_{01} - \underline{a}_{21} = \cancel{0}^{1} - \cancel{1}^{1} = \cancel{-1}^{0}$

$= 1 - 1 + 2 - 4 = -2$

$\sum_k p_k(\underline{a}_{ik} - \underline{a}_{jk}) = (-2)\cdot 0 + (-2)(\cancel{-1}^{0}) = \cancel{2}\ 0$

$\sum_k p_k^2 = 4 + 4 = 8$

$\overline{\alpha} = 0/8 = 0 \leqq 0$

$g(0) = \sum_k(\underline{a}_{0k} - \underline{a}_{2k})^2 = 0 + \cancel{1}^{0} = \cancel{1}^{0},$

$g(\alpha) = 8\alpha^2 + 2\cdot 0\,\alpha + 0 = 8\cdot 10^{-4} \approx 0.0008$

$d_{02}(\alpha) = \sqrt{Q(\alpha)} = \sqrt{g(\alpha)} = \sqrt{0.0008} \fallingdotseq 0.0283$

$al=1$, $\alpha=0.01$の時の$d_{ij}(\alpha)$の値, $ld=2=l$.

$d_{03}(\alpha)=?$

$i \swarrow \; \downarrow j$

$p_0=\underline{a}_{30}-\underline{a}_{00}+\overline{a}_{00}-\overline{a}_{30}$, $\underline{a}_{00}-\underline{a}_{30}=0-0=0$,

$=0-0+1-4=-3$,

$p_1=\underline{a}_{31}-\underline{a}_{01}+\overline{a}_{01}-\overline{a}_{31}$, $\underline{a}_{01}-\underline{a}_{31}=1-2=-2$,

$=2-1+2-3=1-1=0$,

$\sum_k p_k(\underline{a}_{ik}-\underline{a}_{jk})=(-3)\cdot 0+0\cdot(-2)=0$,

$\sum_k p_k^2=9+0=9$

$\overline{\alpha}=0\leqq 0$.

$g(0)=\sum_k(\underline{a}_{0k}-\underline{a}_{3k})^2=(0-0)^2+(1-2)^2=1$,

$g(\alpha)=9\cdot\alpha^2+2\cdot 0\cdot\alpha+1=1.0001$

$d_{03}(\alpha)=\sqrt{R(g)}=\sqrt{g(\alpha)}=\sqrt{1.0001}=1.00005\approx 1.00$ //

$d_{04}(\alpha)=?$

$i \swarrow \; \swarrow j$

$p_0=(\underline{a}_{40}-\underline{a}_{00})+(\overline{a}_{00}-\overline{a}_{40})$, $\underline{a}_{00}-\underline{a}_{40}=0-2=-2$,

$=2-0+1-4=-1$

$p_1=\underline{a}_{41}-\underline{a}_{01}+\overline{a}_{01}-\overline{a}_{41}$, $\underline{a}_{01}-\underline{a}_{41}=1-3=-2$,

$=3-1+2-4=0$

$\sum_k p_k(\underline{a}_{ik}-\underline{a}_{jk})=(-1)(-2)+0(-2)=2$,

$\sum_k p_k^2=1+0=1$,

$\overline{\alpha}=-2/1=-2\leqq 0$.

$g(0)=\sum_k(\underline{a}_{0k}-\underline{a}_{4k})^2=4+4=8$

$g(\alpha)=1\cdot\alpha^2+2\cdot 2\cdot\alpha+8\doteqdot 8+\frac{4}{100}=8.04$

$d_{04}(\alpha)=\sqrt{R(\alpha)}=\sqrt{g(\alpha)}=\sqrt{8.04}\doteqdot 2.835\approx 2.84$

$d_{12}(\alpha) = ?$

$i \swarrow \ \downarrow j$

$p_0 = \underline{a}_{20} - \underline{a}_{10} + \overline{a}_{10} - \overline{a}_{20}$, $\underline{a}_{10} - \underline{a}_{20} = 0 - 0 = 0$,

$= 0 - 0 + 2 - 3 = -1$,

$p_1 = \underline{a}_{21} - \underline{a}_{11} + \overline{a}_{11} - \overline{a}_{21}$, $\underline{a}_{11} - \underline{a}_{21} = 1 - 1 = 0$,

$= 1 - 1 + 3 - 4 = -1$,

$\sum_k p_k(\underline{a}_{ik} - \underline{a}_{jk}) = (-1)\cdot 0 + (-1)\cdot 0 = 0$, $\sum_k p_k^2 = 1 + 1 = 2$

$\overline{\alpha} = 0 \leqq 0$.

$q(0) = \sum_k (\underline{a}_{1k} - \underline{a}_{2k})^2 = 0$,

$q(\alpha) = 2\alpha^2 + 2\cdot 0\cdot \alpha + 0 = 2\alpha^2$

$d_{12}(\alpha) = \sqrt{Q(\alpha)} = \sqrt{q\alpha} = \sqrt{2}\,\alpha = \frac{1.414}{100} = 0.01414$ //

$d_{13}(\alpha) = ?$

$i \swarrow \ \downarrow j$

$p_0 = \underline{a}_{30} - \underline{a}_{10} + \overline{a}_{10} - \overline{a}_{30}$, $\underline{a}_{10} - \underline{a}_{30} = 0 - 0 = 0$,

$= 0 - 0 + 2 - 4 = -2$,

$p_1 = \underline{a}_{31} - \underline{a}_{11} + \overline{a}_{11} - \overline{a}_{31}$, $\underline{a}_{11} - \underline{a}_{31} = 1 - 2 = -1$,

$= 2 - 1 + 3 - 3 = 1$,

$\sum_k p_k(\underline{a}_{ik} - \underline{a}_{jk}) = (-2)\cdot 0 + 1\cdot(-1) = -1$, $\sum p_k^2 = 4 + 1 = 5$

$0 < \overline{\alpha} = 1/5 < \frac{1}{2}$, $0 < \alpha = \frac{1}{100} < \frac{2}{5} = 2\overline{\alpha}$ より

$Q(\alpha) = q(0) = 0^2 + (-1)^2 = 1$,

$d_{13}(\alpha) = \sqrt{q(0)} = \sqrt{1} = 1.00$ //

$\alpha = 0.01,\ ld = 2 = l$

$d_{14}(\alpha) = ?$

$i$ $j$

$p_0 = a_{40} - a_{10} + \bar{a}_{10} - \bar{a}_{40},\quad a_{10} - a_{40} = 0 - 2 = -2,$

$= 2 - 0 + 2 - 4 = 0,$

$p_1 = a_{41} - a_{11} + \bar{a}_{11} - \bar{a}_{41},\quad a_{11} - a_{41} = 1 - 3 = -2,$

$= 3 - 1 + 3 - 4 = 1,$

$\sum_k p_k(a_{ik} - a_{jk}) = 0(-2) + 1\cdot(-2) = -2,$

$\sum_k p_k^2 = 0 + 1 = 1,$

$\bar{\alpha} = -2$ より $g(0) = \sum_k (a_{1k} - a_{4k})^2 = 4 + 4 = 8,$

$\geqq \frac{1}{2}$

$d_{14}(\alpha) = \sqrt{g(0)} = \sqrt{8} = 2.83$ //

$d_{23}(\alpha) = ?$

$i$ $j$

$p_0 = a_{30} - a_{20} + \bar{a}_{20} - \bar{a}_{30},\quad a_{20} - a_{30} = 0 - 0 = 0,$

$= 0 - 0 + 3 - 4 = -1,$

$p_1 = a_{31} - a_{21} + \bar{a}_{21} - \bar{a}_{31},\quad a_{21} - a_{31} = 1 - 2 = -1,$

$= 2 - 1 + 4 - 3 = 2,$

$\sum_k p_k(a_{ik} - a_{jk}) = (-1)\cdot 0 + 2(-1) = -2,$

$\sum p_k^2 = 1 + 4 = 5,\quad g(0) = 0^2 + (-1)^2 = 1,$

$0 < \bar{\alpha} = 2/5 < \frac{1}{2}$ で $0 < \alpha = \frac{1}{100} \leqq 2\bar{\alpha} = \frac{4}{5}$ より

$d_{23}(\alpha) = \sqrt{g(0)} = \sqrt{1} = 1.00$ //

$\alpha = 0.01,\ \ell d = 2 = \ell.$

$d_{24}(\alpha) = ?$

$i \quad j$

$p_0 = (\underline{a}_{40} - \underline{a}_{20}) + (\overline{a}_{20} - \overline{a}_{40}),\quad \underline{a}_{20} - \underline{a}_{40} = 0 - 2 = -2,$

$= 2 - 0 + 3 - 4 = 1$

$p_1 = \underline{a}_{41} - \underline{a}_{21} + \overline{a}_{21} - \overline{a}_{41},\quad \underline{a}_{21} - \underline{a}_{41} = 1 - 3 = -2,$

$= 3 - 1 + 4 - 4 = 2,$

$\sum_k p_k(\underline{a}_{ik} - \underline{a}_{jk}) = 1\cdot(-2) + 2(-2) = -6,\quad \sum_k p_k^2 = 1 + 4 = 5,$

$\overline{\alpha} = 6/5 > \frac{1}{2},\quad g(0) = 4 + 4 = 8,$

$d_{24}(\alpha) = \sqrt{g(0)} = \sqrt{8} = 2.83$ //

$d_{34}(\alpha) = ?$

$i \quad j$

$p_0 = \underline{a}_{40} - \underline{a}_{30} + \overline{a}_{30} - \overline{a}_{40},\quad \underline{a}_{30} - \underline{a}_{40} = 0 - 2 = -2,$

$= 2 - 0 + 4 - 4 = 2,$

$p_1 = \underline{a}_{41} - \underline{a}_{31} + \overline{a}_{31} - \overline{a}_{41},\quad \underline{a}_{31} - \underline{a}_{41} = 2 - 3 = -1,$

$= 3 - 2 + 3 - 4 = 0,$

$\sum_k p_k(\underline{a}_{ik} - \underline{a}_{jk}) = 2(-2) + 0(-1) = -2,\quad \sum p_k^2 = 4,$

$\overline{\alpha} = 2/4 = \frac{1}{2}$ より $g(0) = 4 + 1 = 5$

$d_{34}(\alpha) = \sqrt{g(0)} = \sqrt{5} = 2.24$ //

| i \ j | 1 | 2 (1) | 3 (2) | 4 (3) | 5 (4) |
|---|---|---|---|---|---|
| 1 (0) | | 0.014 | 0.0283 | 1.00 | 2.84 |
| 2 (1) | 0.014 | | 0.014 | 1.00 | 2.83 |
| 3 (2) | 0.0283 | 0.014 | | 1.00 | 2.83 |
| 4 (3) | 1.00 | 1.00 | 1.00 | | 2.24 |
| 5 (4) | 2.28 | 2.83 | 2.83 | 2.24 | |

# $\alpha=0.01$ に対する最適 Clustering，DP計算

• Stage1の各 stateに対するdpvの値はeval表を引けばよい。

• Stage2で

0.007 + 0.5 = 0.507 　 ~~1.84~~ 0.01415 + 0.5 = ~~1.34~~ 0.51415 　 0.5 + 0.007 = 0.507

dpv[0123] = min{ dpv[01]+dpv[23], dpv[02]+dpv[13], dpv[03]+dep[12],

dpv[012], dpv[013], dpv[023], dpv[123] }

~~0.6827~~ 0.0188* 　 0.6713 　 ~~1.34~~ 0.626 　 0.6713

= ~~0.507~~ 0.0188, trace-back[2][0123] → trace-back[2][0] = ~~1~~0. ……⓪

0.007 + 1.415 = 1.422 　 ~~1.84~~ 0.01415 + 1.415 = 1.429 　 1.42 + 0.007 = 1.427

dpv[0124] = min{ dpv[01]+dpv[24], dpv[02]+dpv[14], dpv[04]+dpv[12],

dpv[012], d [014], dpv[024], dpv[124] }

~~0.6827~~ 0.0188* 　 1.8947 　 ~~2.8833~~ 1.8924 　 1.8913

= ~~1.422~~ 0.0188, trace-back[2][0124] → trace-back[2][1] = ~~1~~0. ……①

0.007 + 1.12 = 1.127 　 0.5 + 1.415 = 1.915 　 1.42 + 0.5 = 1.92

dpv[0134] = min{ dpv[01]+dpv[34], dpv[03]+dpv[14], dpv[04]+dpv[13],

dpv[013], dpv[014], dpv[034], dpv[134] }

0.6713* 　 1.8947 　 2.0267 　 2.0233

= 0.6713, trace-back[2][0134] → trace-back[2][2] = 11. ……②

~~1.04~~ 0.01415 + 1.12 = ~~2.13~~ 1.13415 　 0.5 + 1.415 = 1.915 　 1.42 + 0.5 = 1.92

dpv[0234] = min{ dpv[02]+dpv[34], dpv[03]+dpv[24], dpv[04]+dpv[23],

dpv[023], dpv[024], dpv[034], dpv[234] }

~~1.34~~ 0.6276* 　 ~~2.8833~~ 1.8924 　 2.0267 　 2.0233

= ~~1.915~~ 0.676245, trace-back[2][0234] → trace-back[2][3] = ~~1~~3. ……③

0.007 + 1.12 = 1.127 　 0.5 + 1.415 = 1.915 　 1.415 + 0.5 = 1.915

dpv[1234] = min{ dpv[12]+dpv[34], dpv[13]+dpv[24], dpv[14]+dpv[23],

dpv[123], dpv[124], dpv[134], dpv[234] }

0.6713* 　 1.8913 　 2.0233 　 2.0233

= ~~0.6713~~ 0.0188, trace-back[2][1234] → trace-back[2][4] = 16. ……④

Stage3で

min{⓪, ①, ②, ③, ④} = 0.507 at ⓪ より

$\alpha=0.01$ に対する最適 Clusteringは {0,1,2}, {3}, {4} で

目的関数値 = ~~0.507~~ 0.0188 //

OutCluster.txt

Kurano2004-24 N = 18 M = 3

alpha partition

alpha = 0.100000 min. DP value = 66.633392 pantition = 11 9 6 3 1 0 17 12 10 4 2 16 15 14 13 8 7 5

alpha = 0.200000 min. DP value = 66.781929 pantition = 11 9 6 3 1 0 17 12 10 4 2 16 15 14 13 8 7 5

alpha = 0.300000 min. DP value = 66.938713 pantition = 16 15 14 12 8 17 10 5 4 2 13 11 9 7 6 3 1 0

alpha = 0.400000 min. DP value = 67.190918 pantition = 16 15 14 12 8 17 10 5 4 2 13 11 9 7 6 3 1 0

alpha = 0.500000 min. DP value = 67.532219 pantition = 16 15 14 12 8 17 10 5 4 2 13 11 9 7 6 3 1 0

alpha = 0.600000 min. DP value = 67.972733 pantition = 16 15 14 12 8 17 10 5 4 2 13 11 9 7 6 3 1 0

alpha = 0.700000 min. DP value = 68.457413 pantition = 16 15 14 8 5 17 12 10 4 2 13 11 9 7 6 3 1 0

alpha = 0.800000 min. DP value = 69.090225 pantition = 16 15 14 8 5 17 12 10 4 2 13 11 9 7 6 3 1 0

alpha = 0.900000 min. DP value = 69.874237 pantition = 16 15 14 8 5 17 12 10 4 2 13 11 9 7 6 3 1 0

alpha = 1.000000 min. DP value = 70.830887 pantition = 16 15 14 8 5 17 12 10 4 2 13 11 9 7 6 3 1 0

# FMCDP.cpp実行時間実験

$ld=20,\ M=3$

| N | 計算時間 | メモリ |
|---|---|---|
| 16 | 1分54秒 | 130MB |
| 17 | 5分10秒 | 133.6MB |
| 18 | 36分=2123秒 | |
| 19 | | |
| 20 | | |

Tecra 9000　CPU Pentium (R)III Mobile 1.2GHz,
メモリ約1GB

2004.3.4.
岩村メモ

FM-DPの入力ファイル内容

Kurano2004-24 3 24 20

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 8.6 | 9.0 | 5.7 | 8.0 | 6.3 | 8.1 | 5.5 | 7.3 | 6.0 |
| | 7.3 | 5.8 | 7.3 | 5.7 | 8.4 | 2.6 | 6.7 | 4.7 | 6.4 |
| | 8.5 | 9.0 | 6.6 | 8.4 | 6.7 | 8.2 | 6.7 | 8.4 | 5.7 |
| | 7.5 | 5.8 | 8.0 | 6.3 | 8.2 | 6.4 | 8.4 | 6.3 | 9.0 |
| | 6.3 | 9.0 | 7.7 | 9.0 | | | | | |
| 2 | 6.4 | 8.4 | 5.7 | 8.3 | 6.4 | 8.3 | 5.5 | 7.4 | 4.6 |
| | 7.6 | 6.9 | 8.3 | 4.9 | 6.5 | 5.8 | 7.3 | 6.8 | 8.3 |
| | 7.4 | 8.5 | 5.8 | 8.4 | 4.8 | 7.2 | 6.3 | 8.4 | 5.5 |
| | 7.2 | 5.2 | 7.2 | 4.3 | 6.4 | 5.6 | 8.4 | 4.6 | 8.5 |
| | 5.5 | 8.4 | 5.6 | 8.5 | | | | | |
| 3 | 2.5 | 3.3 | 6.5 | 7.4 | 5.6 | 6.4 | 8.4 | 9.0 | 1.5 |
| | 2.3 | 5.4 | 6.0 | 7.5 | 8.3 | 4.4 | 5.0 | 6.4 | 7.1 |
| | 8.3 | 9.0 | 6.4 | 7.3 | 6.3 | 7.1 | 7.4 | 8.3 | 1.0 |
| | 1.3 | 6.4 | 7.3 | 4.5 | 5.3 | 4.6 | 5.3 | 5.5 | 6.2 |
| | 8.4 | 9.0 | 4.5 | 5.3 | | | | | |
| 4 | 4.8 | 5.3 | 7.7 | 8.3 | 5.8 | 6.2 | 7.7 | 8.2 | 5.7 |
| | 6.2 | 5.7 | 6.2 | 7.7 | 8.2 | 6.8 | 7.2 | 7.7 | 8.2 |
| | 6.8 | 7.2 | 7.7 | 9.0 | 6.7 | 7.2 | 7.7 | 8.2 | 5.7 |
| | 6.1 | 7.8 | 9.0 | 4.7 | 5.2 | 5.7 | 6.2 | 3.9 | 5.2 |
| | 4.8 | 6.3 | 5.8 | 6.2 | | | | | |
| 5 | 3.8 | 4.1 | 2.8 | 3.2 | 4.8 | 5.3 | 4.7 | 5.1 | 4.7 |
| | 5.2 | 4.6 | 5.0 | 4.8 | 5.4 | 4.7 | 5.2 | 6.6 | 7.3 |
| | 2.7 | 3.2 | 6.4 | 7.1 | 6.7 | 7.4 | 8.6 | 9.0 | 2.8 |
| | 3.2 | 6.8 | 7.2 | 4.9 | 5.2 | 2.8 | 3.1 | 8.5 | 8.9 |
| | 4.7 | 5.2 | 2.8 | 3.3 | | | | | |
| 6 | 7.1 | 9.0 | 5.0 | 7.9 | 6.0 | 8.0 | 2.0 | 4.9 | 1.0 |
| | 1.9 | 2.9 | 6.0 | 5.9 | 8.0 | 3.2 | 6.9 | 6.1 | 7.9 |
| | 3.2 | 6.0 | 6.0 | 7.9 | 6.5 | 8.8 | 5.0 | 6.7 | 3.8 |
| | 6.0 | 5.0 | 7.8 | 2.1 | 3.9 | 5.1 | 7.1 | 8.0 | 9.0 |
| | 6.1 | 7.8 | 6.1 | 8.0 | | | | | |
| 7 | 6.7 | 9.0 | 8.6 | 9.0 | 6.4 | 8.3 | 8.2 | 9.0 | 5.2 |
| | 8.6 | 8.7 | 9.0 | 8.7 | 9.0 | 8.7 | 9.0 | 8.3 | 9.0 |
| | 7.2 | 9.0 | 7.9 | 9.0 | 7.8 | 9.0 | 7.3 | 9.0 | 6.5 |
| | 8.8 | 5.6 | 8.8 | 7.3 | 9.0 | 6.7 | 9.0 | 2.7 | 5.2 |
| | 7.9 | 9.0 | 8.3 | 9.0 | | | | | |